



“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

数据库应用

(Visual FoxPro 9.0)

王 苒 主 编

王 璐 金雯岗 陆红彬 副主编

王 健 主 审



電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书根据教育部颁发的《中等职业学校专业教学标准（试行）信息技术类（第一辑）》中的相关教学内容和要求编写。本书的编写从满足经济发展对高素质劳动者和技能型人才的需求出发，在课程结构、教学内容、教学方法等方面进行了新的探索与改革创新，以利于学生更好地掌握本课程的内容，利于学生理论知识的掌握和实际操作技能的提高。

本书遵照知识模块体系，同时结合数据库项目开发的工作流程，以“学生成绩管理信息系统”作为指导教学的项目开发实例，以“图书管理信息系统”作为课后拓展练习项目，将整个教学内容划分成5个单元，通过16个项目、53个任务，系统介绍了数据库需求分析及设计、数据库及表的实现、数据查询和视图的操作、程序设计基础、系统设计与实现等内容。

本书是软件与信息服务专业核心课程教材，也可供各种数据库开发培训课程作为教材使用，也适合需要提高自己计算机应用技能的广大计算机爱好者使用。

本书配有教学指南、电子教案和案例素材，详见前言。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

数据库应用. Visual FoxPro 9.0 / 王苒主编. —北京：电子工业出版社，2017.8

ISBN 978-7-121-24893-1

数... 王... 关系数据库系统—中等专业学校—教材 TP311.138

中国版本图书馆 CIP 数据核字（2014）第 274963 号

策划编辑：关雅莉

责任编辑：柴 灿

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：15.75 字数：463 千字

版 次：2017 年 8 月第 1 版

印 次：2017 年 8 月第 1 次印刷

定 价：34.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254617，luomn@phei.com.cn。

编审委员会名单

主任委员：

武马群

副主任委员：

王 健 韩立凡 何文生

委 员：

丁文慧	丁爱萍	于志博	马广月	马之云	马永芳	马玥桓	王 帅	王 苒
王 彬	王晓姝	王家青	王皓轩	王新萍	方 伟	方松林	孔祥华	龙天才
龙凯明	卢华东	由相宁	史宪美	史晓云	冯理明	冯雪燕	毕建伟	朱文娟
朱海波	向 华	刘 凌	刘小华	刘天真	关 莹	江永春	许昭霞	孙宏仪
苏曰太夫	杜 琚	杜宏志	杜秋磊	李 飞	李 娜	李华平	李宇鹏	杨 杰
杨 怡	杨春红	吴 伦	何 琳	余运祥	邹贵财	沈大林	宋 微	张 平
张 侨	张 玲	张士忠	张文库	张东义	张兴华	张呈江	张建文	张凌杰
张媛媛	陆 沁	陈 玲	陈 颜	陈丁君	陈天翔	陈观诚	陈佳玉	陈泓吉
陈学平	陈道斌	范铭慧	罗 丹	周 鹤	周海峰	庞 震	赵艳莉	赵晨阳
赵增敏	郝俊华	胡 尹	钟 勤	段 欣	段 标	姜全生	钱 峰	徐 宁
徐 兵	高 强	高 静	郭 荔	郭立红	郭朝勇	涂铁军	黄 彦	黄汉军
黄洪杰	崔长华	崔建成	梁 姗	彭仲昆	葛艳玲	董新春	韩雪涛	韩新洲
曾平驿	曾祥民	温 晞	谢世森	赖福生	谭建伟	戴建耘	魏茂林	

序 | PROLOGUE

当今是一个信息技术主宰的时代，以计算机应用为核心的信息技术已经渗透到人类活动的各个领域，彻底改变着人类传统的生产、工作、学习、交往、生活和思维方式。和语言和数学等能力一样，信息技术应用能力也已成为人们必须掌握的、最为重要的基本能力。职业教育作为国民教育体系和人力资源开发的重要组成部分，信息技术应用能力和计算机相关专业领域专项应用能力的培养，始终是职业教育培养多样化人才，传承技术技能，促进就业创业的重要载体和主要内容。

信息技术的发展，特别是数字媒体、互联网、移动通信等技术的普及应用，使信息技术的应用形态和领域都发生了重大的变化。第一，计算机技术的使用扩展至前所未有的程度，桌面电脑和移动终端（智能手机、平板电脑等）的普及，网络和移动通信技术的发展，使信息的获取、呈现与处理无处不在，人类社会生产、生活的诸多领域已无法脱离信息技术的支持而独立进行。第二，信息媒体处理的数字化衍生出新的信息技术应用领域，如数字影像、计算机平面设计、计算机动漫游戏、虚拟现实等。第三，信息技术与其他业务的应用有机地结合，如与商业、金融、交通、物流、加工制造、工业设计、广告传媒、影视娱乐等结合，形成了一些独立的生态体系，综合信息处理、数据分析、智能控制、媒体创意、网络传播等日益成为当前信息技术的主要应用领域，并诞生了云计算、物联网、大数据、3D 打印等指引未来信息技术应用的发展方向。

信息技术的不断推陈出新及应用领域的综合化和普及化，直接影响着技术、技能型人才的培养定位，并引领着职业教育领域信息技术或计算机相关专业与课程改革、配套教材的建设，使之不断推陈出新、与时俱进。

2009 年，教育部颁布了《中等职业学校计算机应用基础大纲》，2014 年，教育部在 2010 年新修订的专业目录基础上，相继颁布了“计算机应用、数字媒体技术应用、计算机平面设计、计算机动漫与游戏制作、计算机网络技术、网站建设与管理、软件与信息服务、客户信息服务、计算机速录”等 9 个信息技术类相关专业的教学标准，确定了教学实施及核心课程内容的指导意见。本书就是以此为依据，结合当前最新的信息技术发展形势和企业应用案例组织开发和编写的。



本套系列教材的主要特色

● 对计算机专业类相关课程的教学内容进行重新整合

本套教材面向学生的基础应用能力，设定了系统操作、文档编辑、网络使用、数据分析、媒体处理、信息交互、外设与移动设备应用、系统维护维修、综合业务运用等内容；针对专业应用能力，根据专业和职业能力方向的不同，结合企业的具体应用业务规划了教材内容。

● 以岗位工作过程来确定学习任务和目标，综合提升学生的专业能力、过程能力和职位差异能力

本套教材通过工作过程为导向的教学模式和模块化的知识能力整合结构，体现产业需求与专业设置、职业标准与课程内容、生产过程与教学过程、职业资格证书与学历证书、终身学习与职业教育的“五对接”。从学习目标到内容的设计上，本套教材不再仅仅是专业理论内容的复制，而是经由职业岗位实践——工作过程与岗位能力分析——技能知识学习应用内化的学习实训导引和案例。借助知识的重组与技能的强化，达到企业岗位情境和教学内容要求相贯通的课程融合目标。

● 以项目教学和任务案例实训作为主线

本套教材通过项目教学，构建了工作业务的完整流程和岗位能力需求体系。项目的确定应遵循3个基本目标：核心能力的熟练程度、技术更新与延伸的再学习能力、不同业务情境应用的适应性。教材借助以校企合作为基础的实训任务，以应用能力为核心、以案例为线索，通过设立情境、任务解析、引导示范、基础练习、难点解析与知识延伸、能力提升训练和总结评价等环节引领学者在任务的完成过程中积累技能、学习知识，并迁移到不同业务情境的任务解决过程中，使学者在未来可以从容面对不同应用场景的工作岗位。

当前，全国职业教育领域都在深入贯彻全国工作会议精神，学习领会中央领导对职业教育的重要批示，全力加快推进现代职业教育。国务院出台的《加快发展现代职业教育的决定》明确提出要“形成适应发展需求、产教深度融合、中职高职衔接、职业教育与普通教育相互沟通，体现终身教育理念，具有中国特色、世界水平的现代职业教育体系”。现代职业教育体系的建立将带来人才培养模式、教育教学方式和办学体制机制的巨大变革，这无疑给职业院校信息技术应用人才培养提出了新的目标。计算机类相关专业的教学必须要适应改革，始终把握技术发展和技术技能人才培养的最新动向，坚持产教融合、校企合作、工学结合、知行合一，为培养出更多适应产业升级转型和经济发展的高素质职业人才做出更大贡献！

前言 | PREFACE

为建立健全教育质量保障体系，提高职业教育质量，教育部于 2014 年颁布了中等职业学校专业教学标准（以下简称专业教学标准）。专业教学标准是指导和管理中等职业学校教学工作的主要依据，是保证教育教学质量和人才培养规格的纲领性教学文件。在“教育部办公厅关于公布首批《中等职业学校专业教学标准（试行）》目录的通知”（教职成厅[2014]11 号文）中，强调“专业教学标准是开展专业教学的基本文件，是明确培养目标和规格、组织实施教学、规范教学管理、加强专业建设、开发教材和学习资源的基本依据，是评估教育教学质量的主要标尺，同时也是社会用人单位选用中等职业学校毕业生的重要参考。”

本书特色

本书根据教育部颁发的《中等职业学校专业教学标准（试行）信息技术类（第一辑）》中的相关教学内容和要求编写。

本书以数据库初学者为教学对象，在内容上以项目的形式，按照系统开发生命周期有序组织教学内容，设计了“学生成绩管理信息系统”作为教学主线，同时在课后拓展时，以设计并实现“图书管理信息系统”作为巩固练习，使学生逐步达到知识和技能的教学目标。

全书将实现一个数据库应用系统作为一个大项目，共分为 5 个单元，每个单元包括若干个拆分后的小项目，每项目都包含若干具体的实施任务。在教材体例设计上，每个任务根据需要包含了任务目标、任务描述、相关知识、任务分析、实施步骤、问题探究六部分，帮助学生明确目标，知识迁移、提高分析问题和解决问题的能力；每项目前给出知识目标和技能目标，项目后根据具体需要包含阅读材料、项目小结、上机实习和知识巩固四部分，帮助学生课后拓展、巩固知识，强化技能。

在数据库应用基础教学内容上，以 Visual FoxPro 9.0 作为系统环境，完成本书中所有数据库对象及应用程序的创建、编辑、修改和运行。全书内容分为两大部分：数据库的需求分析及设计（即数据处理及管理，包括前三个单元）和数据库应用系统的设计与实现（即数据库程序设计，包括后两个单元），这五个单元分别如下。

单元一：数据库需求分析及设计，介绍关系数据库概念模型的设计过程及方法。

单元二：数据库及表的实现，介绍使用 Visual FoxPro 数据库管理系统实现并维护关系数据库模型的方法和过程。

单元三：数据查询及视图的操作，介绍使用查询设计器、视图设计器创建查询和视图，检索和更新数据的方法和过程，同时介绍了结构化查询语言 SQL 中 SELECT 查询语句的使用方法。

单元四：程序设计基础，介绍在 Visual FoxPro 语言环境中进行结构化程序设计的思想。

单元五：系统设计与实现，模块化设计并实现“学生成绩管理信息系统”，着重介绍面向对象程序设计的思想和方法。

本书是软件与信息服务专业核心课程教材，可供各种数据库开发培训课程作为教材使用，也适合需要提高自己计算机应用技能的广大计算机爱好者使用。

本书作者

本书由王苒主编，王璐、金雯岗、陆红彬副主编，王健主审，由于编者水平有限，书中难免存在疏漏之处，敬请广大读者批评指正。

教学资源

为了提高学习效率和教学效果，方便教师教学，作者为本书配备包括电子教案、教学指南、素材文件以及习题参考答案等配套的教学资源。请有此需要的读者登录华信教育资源网（<http://www.hxedu.com.cn>）免费注册后进行下载，有问题时请在网站留言板留言或与电子工业出版社联系（E-mail: hxedu@phei.com.cn）。

编 者



CONTENTS | 目录



单元一 数据库需求分析及设计

项目一 构建关系数据库	2
任务一 建立“学生成绩管理信息系统”关系模型	2
任务二 设计“学生成绩”数据库	5
项目二 认识数据库管理系统 Visual FoxPro	15
任务一 安装 Visual FoxPro 9.0	15
任务二 使用 Visual FoxPro 用户界面	16
任务三 在“命令”对话框中执行简单命令	19
任务四 创建简单表达式	22
任务五 调用常用函数	29
任务六 使用项目管理器管理文件	32

单元二 数据库及表的实现

项目一 创建学生信息表	45
任务一 使用表向导创建“学生自然情况表”	45
任务二 使用表设计器创建“课程情况表”	49
任务三 显示和修改“课程情况表”的结构	51
项目二 创建“学生成绩管理”数据库	58
任务一 创建数据库	58
任务二 添加自由表	61
任务三 新建数据库表	64
任务四 修改数据库	70
项目三 管理学生信息表数据	75
任务一 输入和追加表数据	75



任务二	浏览表数据	79
任务三	修改和更新表数据	82
任务四	删除和恢复表数据	85
任务五	学生信息排序	87
任务六	检索学生数据	91
任务七	多表的操作	93

单元三 数据查询和视图的操作

项目一	使用查询设计器	109
任务一	单表查询学生基本情况	109
任务二	多表查询学生选课情况	114
任务三	创建交叉表, 查询学生各门课程的成绩	117
项目二	使用 SQL 命令查询	120
任务一	单表条件查询“选修课程情况”	120
任务二	多表关联查询“成绩不及格的学生信息”	123
任务三	条件查询“计算机网络课程成绩优秀的女学生”	125
任务四	分组查询、统计选课 3 门以上学生的平均成绩和选课门数	127
任务五	查询“计算机应用专业全体男生情况”, 按出生日期降序排序	129
项目三	使用视图更新数据表	132
任务一	创建本地视图	132
任务二	使用视图更新数据	134
任务三	创建和运行远程视图	136

单元四 程序设计基础

项目一	编写程序输出指定学号的学生信息	144
任务一	创建、调试并运行程序	144
任务二	从键盘输入学号, 查询并显示	149
任务三	格式化输出学生信息	151
任务四	如果学号不存在, 输出提示信息	153
任务五	判断学生成绩, 并格式化输出成绩等级	155
任务六	格式化输出所有学生的信息	158
项目二	模块化编写“输出指定学号的学生信息”程序	170
任务一	编写“格式化显示学生信息”子程序, 并在主程序中调用	170
任务二	将子程序改写为过程文件, 并在主程序中调用	173
任务三	将获取学生成绩的过程文件“GetCj”转换成函数文件	175

单元五 系统设计与实现

项目一 设计并制作“学生成绩管理信息系统”的主表单	184
任务 设计“学生成绩管理信息系统”的主界面	184
项目二 设计并制作“学生成绩管理信息系统”的登录界面	191
任务一 设计登录界面的背景图像	191
任务二 设计登录界面的“用户名”和“密码”	192
任务三 设计登录界面的“确定”和“取消”按钮	195
项目三 创建“学生成绩管理系统”主表单菜单	202
任务 设计“学生成绩管理系统”的主表单菜单	202
项目四 实现学生成绩信息维护和查询功能	209
任务一 实现“学生信息维护”功能	209
任务二 实现“学生信息查询”功能	219
项目五 创建学生信息报表	228
任务 设计并制作“学生管理”中的“学生信息报表”	228
项目六 发布“学生成绩管理信息系统”	235
任务一 连编“学生成绩管理信息系统”	235
任务二 发布“学生成绩管理信息系统”	236



单元一

数据库需求分析及设计

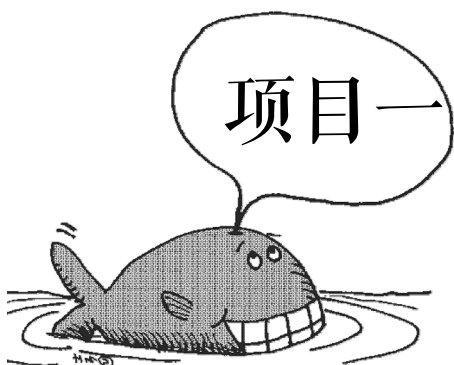
知识目标

- 1) 了解数据库管理技术的发展过程。
- 2) 区分数据库、数据库管理系统、数据库应用系统、数据库系统的概念。
- 3) 了解 3 种数据模型，掌握关系数据模型的概念、构建方法、构建步骤。
- 4) 掌握关系数据库设计的方法及步骤。
- 5) 掌握关系、元组、记录、属性、字段、域、主关键字、外部关键字的概念。

技能目标

- 1) 能够根据用户需求确定关系数据模型，并使用 E-R 模型图描述。
- 2) 能够根据概念模型设计关系数据库。
- 3) 能运用关系数据库设计原则优化表、字段及表间的关系。





构建关系数据库

项目说明

早在计算机出现之前，学校对学生成绩信息的管理主要通过手工录入、纸质保存的方式，这种方式工作效率低、错误率高、数据有冗余，且如果保存不当，容易造成数据丢失。

随着计算机的出现和普及，其强大的运算能力被运用到各应用领域。这时学校开发出第一版本的“学生成绩管理信息系统”，该程序与学生成绩的所有数据相互关联，但是这些学生成绩数据不能被学生科的“学生学籍管理信息系统”或者财务科的“学生经费管理信息系统”等其他相关系统使用，数据的共享性差。

随着数据库管理系统的出现和发展，关系模型被几乎所有的数据库管理系统支持，Visual FoxPro（简称 VFP）就是一种关系数据库管理系统。它的主要特点是数据与程序分离并能被其他应用程序共享。因此，“学生成绩管理信息系统”又需要按照关系模型重新进行设计。本项目将以本系统为例，引导大家学习如何构建“学生成绩管理信息系统”的关系数据库。

本项目分以下两步实施。

任务一：建立“学生成绩管理信息系统”关系模型。

任务二：设计“学生成绩”数据库。

任务一 建立“学生成绩管理信息系统”关系模型

任务目标

能根据用户需求分析组成系统的实体及实体间的联系，并建立关系概念模型。

能使用 E-R 模型图描述概念模型。

相关知识

在现实世界中事物之间存在着联系，这些联系是客观存在的，并且是由事物本身的性质所

决定的。

1. 对实体的描述

实体是客观存在并相互区别的事物。实体可以是实际事物，如篮球、学生、图书、商品等，也可以是抽象事件，如进货、选课、借阅等。

属性是实体的特性。例如，篮球实体可以用品牌、型号、出厂日期、颜色、价格来描述；学生可以用学号、姓名、性别、出生日期、入校成绩、专业等属性来描述。属性有“型”和“值”之分，“型”是属性名，“值”为属性的具体内容。例如，“性别”为属性名，“女”为属性值。

实体型是若干属性的型所组成的集合，可以用实体名和属性名集合来表示实体型。例如，学生（学号，姓名，性别，出生日期，入校成绩，专业）代表了一个学生实体型；篮球（品牌，型号，出厂日期，颜色）代表了一个篮球实体型；选课（学生编号，课程编号，选课学期，成绩）代表了一个选课实体型。

实体集是同类型的实体的集合。例如，在学生实体集中，（131401，马克，男，19961214，456，计算机网络）则具体代表了一个学生实体；（131402，汤可欣，女，19960512，564，计算机网络）则又代表了该实体集中另一个学生实体。

2. 实体间的联系及联系的种类

实体之间的对应关系称为联系，反映现实世界事物之间的相互关联。例如，一名学生可以选择多门课程，一门课程可以被多个学生选修；一位读者可以借阅若干本图书，同一本图书可以相继被几个读者借阅。

实体间的联系可以归结为以下 3 种类型。

（1）一对一联系

实体集 A 中的一个实体最多与实体集 B 中的一个实体相对应，反之，实体集 B 中的一个实体最多对应于实体集 A 中的一个实体，则称为实体集 A 与实体集 B 为一对一联系（1—1），如图 1-1 所示。

（2）一对多联系

实体集 A 中的一个实体与实体集 B 中的 n ($n>1$) 个实体相对应，反之，实体集 B 中的一个实体最多与实体集 A 中的一个实体相对应，称为实体间的一对多联系（1— N ），如图 1-2 所示。

（3）多对多联系

实体集 A 中的一个实体与实体集 B 中的 n ($n>1$) 个实体相对应，反之，实体集 B 中的一个实体与实体集 A 中的 m ($m>1$) 个实体相对应，称为实体间的多对多联系（ M — N ），如图 1-3 所示。

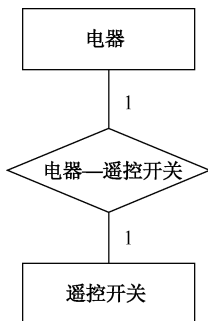


图 1-1 实体 1—1 联系

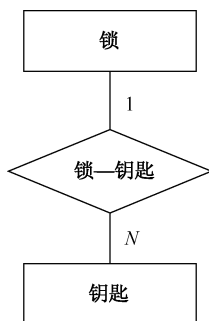


图 1-2 实体 1— N 联系

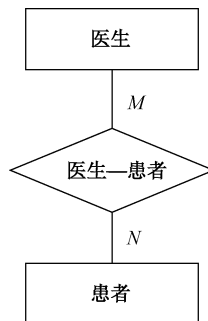


图 1-3 实体 M — N 联系



实施步骤

1. 根据用户需求描述系统功能

学生成绩管理信息系统主要包括 3 个模块：学生信息管理模块、课程信息管理模块和成绩信息管理模块，各模块的具体功能如下。

学生信息管理模块的主要功能包括学生信息的添加、删除、查询、修改和统计。该模块记录学生的学号、姓名、出生日期、性别、专业、年龄、家庭住址、籍贯、父亲、母亲、家庭电话、照片等基本信息。

课程信息管理模块的主要功能包括课程信息的添加、删除、查询、修改和统计。该模块记录课程编号、课程名称、学分、授课学期、学时、考试或考查等信息。

成绩信息管理模块的主要功能包括成绩信息的添加、删除、查询、修改和统计。该模块记录课程编号、学号、选修学期、期中成绩、期末成绩、总成绩、所得学分等基本信息。

2. 主要数据分析

经过上述系统功能分析和需求总结，设计实体及实体关系如下。

(1) 学生实体

学生 (学号, 姓名, 出生日期, 性别, 专业, 籍贯, 照片, 父亲, 母亲, 联系电话, 家庭住址)。

(2) 课程实体

课程 (课程编号, 课程名称, 学分, 授课学期, 授课老师, 学时, 考试)。

(3) 学生选课实体

学生选课 (课程编号, 学号, 授课学期, 期中成绩, 期末成绩, 总成绩, 所得学分)。

学生实体通过选课实体与课程实体发生联系，且为多对多联系。

3. 描述关系模型

使用 E-R (实体-关系) 模型图描述 “学生成绩管理信息系统” 关系模型，如图 1-4 所示。

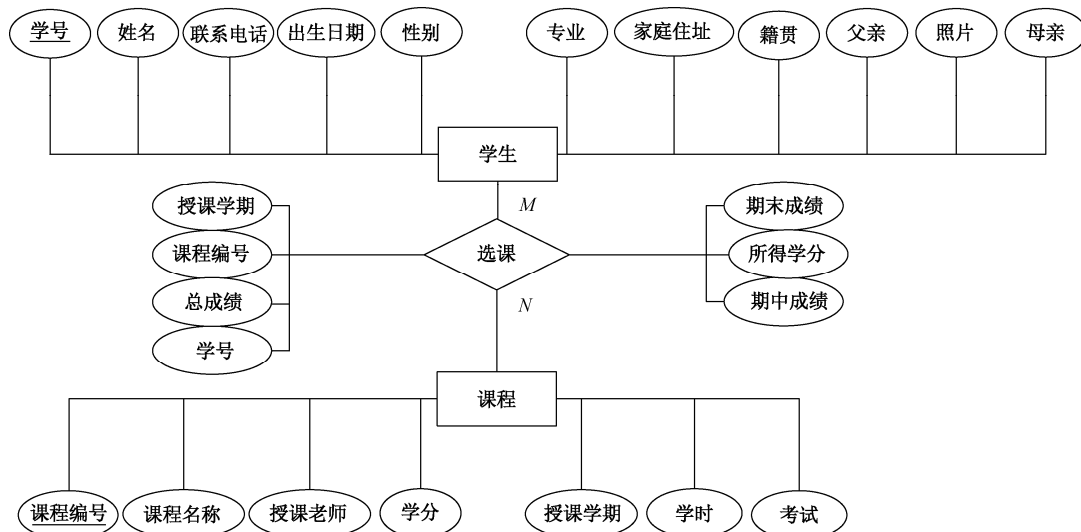


图 1-4 “学生成绩管理信息系统” 关系模型图

任务二 设计“学生成绩”数据库

任务目标

能根据概念模型设计数据物理模型。

能够对关系数据库进行关系运算,对数据进行分析,确定数据表、字段、关系并优化设计。

相关知识

1. 数据库的定义

数据库(以下所有“数据库”均指关系数据库)是按照某种数据模型组织、存储和管理数据的仓库,即数据库是存储在一起的相关数据的集合。这些数据按结构化组织,尽量减少冗余,独立于应用程序进行存储且对数据的增加、删除、修改和检索可以按照一种公用的和可控制的方式进行。

2. 数据库的组成

一个数据库文件代表一个实际的关系模型,它包括多个按照不同用途和分类组织在一起的数据表。每个数据表在关系模型中被称为关系,它代表一个实体集或实体联系的集合,描述为:关系名(属性1,属性2,属性3……),它可以设计成一个数据表。

关系是规范的二维表,因此数据表由若干行和列组成,每一列代表实体的一个属性,它在数据表中称为字段,每一字段的取值范围称为域。每一行代表一个具体实体,它在数据表中称为记录(在关系中称为元组)。

能够唯一标识一条记录的字段被称为主关键字,即在一个实际的关系模型中,实体的主键是表示该实体的数据表的主关键字。例如,在图书管理信息系统中建立图书借阅数据库,该数据库包括图书、读者、读者借阅图书3个实体,因此可以建立3个数据表。其中图书(编号,书名,所属分类,作者,ISBN号,出版社,版本,购买日期,购买价格)可以被设计成图书表,该表的表现形式类似于一个由行和列组成的 Excel 表,每一列代表一个属性,每一行代表一条记录,如图 1-5 所示。

1	2	3	4	5	6	7	8	9
编号	书名	所属分类	作者	ISBN号	出版社	版本	购买日期	购买价格
1	C语言程序设计	机电	徐磊	78902411	电子工业出版社	第一版	2010/12/8	12.5
2	计算机原理	计算机	李道儒	78905231	机械工业出版社	第二版	2013/1/10	0.6
3	机械基础	机电	田国芳	78965325	清华大学出版社	第一版	2012/2/13	25
4	自动控制	机电	曹洪明	789656421	北京大学出版社	第二版	2014/11/5	36.0
5	逻辑电路	电子	李步微	78985412	电子工业出版社	第三版	2009/3/15	13.5
6	电工电子	电子	何晓化	78956320	科学出版社	第一版	2005/12/12	20.0
7	局域网搭建	机电	宁司	78985741	电子工业出版社	第一版	2014/1/25	34.0

图 1-5 图书表

任务分析

设计数据库的基本步骤如下。

1. 设计模型

进行用户需求分析,设计关系模型(此步骤已在任务一中完成)。

2. 规划建表

由关系模型中的实体及实体间的联系规划该数据库中需要建立的表。

定义数据库中的表是数据库设计过程中技巧性最强的一步。要从系统功能中仔细研究需要



从数据库中取出的信息,遵从概念单一化的原则,即一个表描述一个实体或实体间的一种联系,把这些信息分成各种基本实体。例如,在超市货品管理系统中,把货物、进货、出库、报损等每个实体设计成一个独立的表。在销售管理信息系统中,将客户、职工、商品、订单、供应商等每个实体设计成一个独立的表。

3. 确定表中需要的字段

确定表中字段时需要注意以下 3 个问题。

(1) 每个字段直接和表的实体相关

字段代表该实体的属性,不属于该实体的属性不应属于本表。例如,“身高”字段不属于商品表。如果多个表中有重复内容的字段,应删除不必要的字段。例如,图书表中有“图书编号”、“图书名”、“图书分类”等字段,在借阅表中又包含了这些字段,这时,应该删除借阅表中的该类字段(除“图书编号”字段外),保留原始实体中的相同字段。

(2) 以最小的逻辑单位存储信息

如果一个字段中包含了多种数据,则将会很难获取单独的数据,应尽量把信息分解成比较小的逻辑单位。例如,在“土豆食品蔬菜”这个字段值中包含了商品名、商品类别和商品描述 3 项内容,应创建 3 个不同的字段。

(3) 表中的字段应该是原始数据

通常情况下,对于可推导或计算得到的数据,最好不作为字段出现。例如,学生表中有“出生年月”字段,就不必再设计“年龄”字段了。

4. 明确有唯一值的字段——主关键字字段

主关键字字段必须能唯一标识一个具体的实体。为使关系型数据库管理系统能够迅速查询所要信息,数据库的每一个数据表都必须有一个或一组字段,通过该字段的值能够唯一确定存储在表中的记录。这个字段或这组字段被称为主关键字。主关键字字段中的值不能为空值或有重复值。例如,客户表中的“客户编码”、职工表中的“职工号”、商品表中的“商品号”、订单表中的“订单号”字段都是主关键字字段,这些关键字段在 E-R 模型中是带下画线的属性。

5. 确定表之间的关系

各数据表之间的关系代表了各实体间的联系。要建立两个表的关系,可以把其中一个表的主关键字添加到另一个表中,使两个表都具有相同字段。为表示一对多、多对多及一对一 3 种实体联系,具体方法如下。

(1) 一对多联系

现象:表 A 的一个记录在表 B 中可以有多个记录与之对应,但表 B 中的一个记录最多只能和表 A 中的一条记录对应。

方法:将表 A 中的主关键字字段添加到表 B 中,作为表 B 的普通索引关键字,如图 1-6 所示。

(2) 多对多联系

现象:表 A 的一个记录在表 B 中可以对应多个记录,而表 B 中的一个记录在表 A 中也可以对应多个记录。

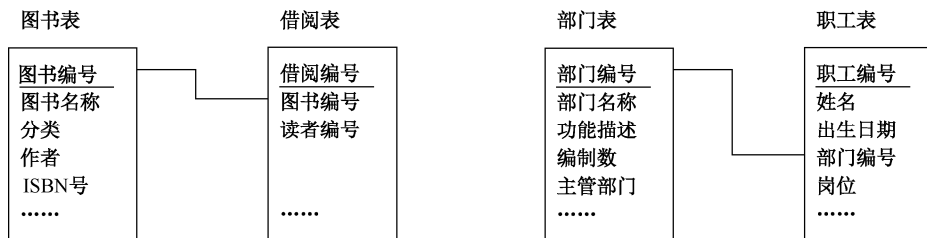


图 1-6 数据表之间一对多联系

方法：创建表 C，将多对多的联系分解成两个一对多联系。新创建的表 C 同时包含表 A 和表 B 的主关键字，分别作为普通索引关键字，同时表 C 的主关键字可以另外设计或将表 A 和表 B 的主关键字作为组合关键字指定为主关键字，如图 1-7 所示。

（3）一对一联系

现象：表 A 的一个记录在表 B 中至多对应一个记录，而表 B 中的一个记录在表 A 中也只对应一个记录。

方法：如果表 A 和表 B 属于同一实体型，则考虑是否合并成一个表。例如，某商场销售人员和员工都是职工，这时可以将销售人员表和职工表合并成一个表 C，并在表 C 中添加字段“岗位”区分职工的类型。

如果表 A 和表 B 不属于同一实体型，则根据具体关系选择其中一个表 A，将它的主关键字放到表 B 中作为外部关键字，也可表示一对一联系。

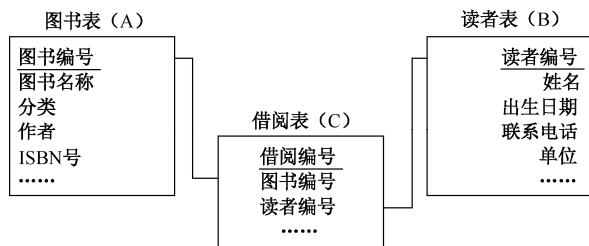


图 1-7 数据表之间多对多联系

6. 优化设计

通过以上各步骤设计出数据库、数据表、字段和联系后，应该从总体上反复推敲设计方案，检查可能存在的缺陷和需要改进的地方，这些缺陷可能会使程序功能难以实现或数据难以维护和使用。

7. 输入具体数据

如果认为表结构已经比较合理，即可继续进行操作并且在表中添加所有初始化数据。



实施步骤

1. 设计模型

确定“学生成绩管理信息系统”概念模型，如图 1-4 所示。

2. 规划建表

确定“学生成绩管理信息系统”数据库及数据表。

建立数据库 StudentDB，该数据库中包含 3 个基本数据表，它们分别如下。

- 学生数据表 StudentDBF，表中包含学生各方面的信息数据。
- 课程数据表 CourseDBF，表中包含课程各方面的信息数据。
- 选修数据表 SelectDBF，表中包含学生选修课程及成绩等方面的信息数据。



3. 确定表中需要的字段

根据学生实体属性确定学生数据表 StudentDBF 的字段。

StudentDBF (学号, 姓名, 出生日期, 性别, 专业, 籍贯, 照片, 父亲、母亲、联系电话, 家庭住址)。

根据课程实体属性确定课程数据表 CourseDBF 的字段。

CourseDBF (课程编号, 课程名称, 学分, 授课学期, 授课老师, 学时, 考试)。

根据选课实体属性确定选修数据表 SelectDBF 的字段。

SelectDBF (课程编号, 学号, 授课学期, 期中成绩, 期末成绩, 总成绩, 所得学分)。

4. 确定主关键字字段

在 E-R 模型图中, 主关键字字段表示为有下画线的属性。主关键字字段是能够唯一标识一个记录的字段。在学生数据表中, “学号” 能唯一标识一个学生, 为主关键字字段; 在课程数据表中, “课程编号” 能唯一标识一门课程, 为该表的主关键字字段; 在选修数据表中, “课程编号” 为主关键字字段。

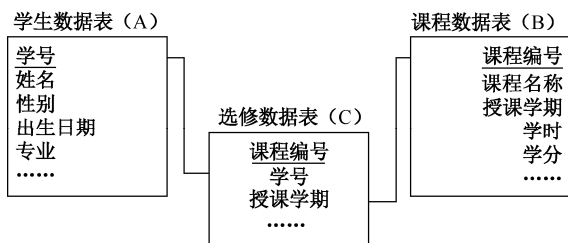


图 1-8 “学生成绩管理信息系统” 中数据表多对多联系

5. 确定表之间的关系

在学生数据表与课程数据表之间, 一名学生可以选择多门课程, 同一门课程可以被多个学生选修, 所以它们之间的联系应为多对多联系。在表的实现关系中, 表现为它们分别和第三个表——“选修表” 之间建立起的一对多联系, 如图 1-8 所示。

6. 优化设计

经过分析检查发现如下情况。

在课程数据表 CourseDBF 中具有“授课学期”字段, 相同字段还在选修数据表 SelectDBF 中出现, 这种情况可能导致数据冗余, 因此需要根据实际情况做出保留和删除。在中等职业学校教学进程中, 学生选修课程是由教学计划确定的, 因此“授课学期”这一属性应归属于课程实体, 放在课程数据表 CourseDBF 中, 在选修表中应把该字段删除。但在大学阶段, 学生有权决定在哪个学期选修哪门课程, 所以该字段最好放在选修数据表 SelectDBF 中。不同的属性最好根据其归属决定它属于哪个数据表。

在选修数据表 SelectDBF 中包含了 3 个成绩字段, 其中“总成绩”可以通过“期中成绩”和“期末成绩”计算出来, 属于冗余字段, 所以删除“总成绩”字段。

7. 输入具体数据

在表中添加具体数据。



问题探究

1. 如何区分主关键字、外部关键字、普通索引关键字?

主关键字是数据表的唯一标识属性, 不能重复, 否则不能准确定义某一条记录。例如, “编

号”可以作为主关键字，“姓名”不可以。

外关键字是指该字段数据来源于另一个表的主关键字，是用来建立两个表之间关联的字段。例如，图书表中的“图书编号”是该表的主关键字，它是借阅表的外关键字。

普通索引关键字是对记录进行排序时采用的一种排序方案，按照该字段值进行升序或降序排序，字段值可以重复。

2. 如何做好优化设计？

【例 1-1】在图书管理信息系统数据库中，图书数据表 BookDBF（图书编号，书名，数量，价格，存放位置，ISBN 号，作者）。

【例 1-2】在招生录取信息系统数据库中，学生填报志愿表 ZYDBF（准考证号，艺术类学校，艺术专业 1，艺术专业 2，1 批学校，1 批专业 1，1 批专业 2，2 批学校……）。

【例 1-3】在订单管理信息系统数据库中，产品数据表 ProductDBF（编号，品名，单位，单价，数量……）；客户数据表 CustomDBF（编号，单位，联系人，地址……）；订单数据表 OrderDBF（订单编号，商品编号，客户编号，下单日期，商品数量，单价，订单总额）。

【例 1-4】在综合人事管理信息系统数据库中，部门表 DepartmentDBF（部门编号，名称，描述，上级编号）；员工表 EmployeeDBF（姓名，性别，民族，出生日期，政治面貌，文化程度，婚姻，籍贯……）；考勤表 DutyDBF（考勤月份，员工编号，全勤天数，出差天数，病假天数，事假天数，矿工天数，法定假天数，年休假天数，迟到时间，早退时间，加班时间……）；考评表 EvaluateDBF（考评月份，员工编号，总体评价，奖励金额，处罚金额，备注……）。

在数据库优化设计中，需要检查以下几个方面。

是否遗漏字段？

是否存在大量空白内容的字段？

是否有包含相同字段的表？

是否有不属于该实体的字段？

是否一个字段内容涵义太多，不适合查询？

是否有合适的主关键字？

是否字段很多记录很少？

例 1-1 中，字段信息不全，添加“图书分类”字段会便于检索。

例 1-2 中，“艺术类学校”和“艺术专业”字段被填写的概率会很小，放在这张表中会浪费磁盘空间，应将报艺术类、师范类、军事院校志愿的另做一张表。

例 1-3 中，产品数据表和客户数据表中共有“单位”字段，产品数据表和订单数据表中共有“单价”字段，删除重复字段。

例 1-4 中，员工表中以“姓名”作为主关键字不合适，因为姓名有重复现象，不能唯一标识一条记录，应该添加“员工编号”字段。



阅读材料

1. 数据库管理技术的发展

数据库管理技术是对数据进行收集、整理、组织、存储、检索、维护等数据处理的技术，是数据处理的核心。

计算机数据管理技术主要经历了人工管理、文件系统管理、数据库系统管理阶段。

人工管理阶段（20 世纪 50 年代以前）如图 1-9 所示，其特点如下。



- 数据不独立。没有专门管理数据的软件，数据由计算或处理它的程序自行携带。
- 数据不能长期保存。早期计算机硬件还没有外磁盘存储器等存储设备，数据在程序运行结束后和程序一起从内存中释放。
- 数据不能共享。没有对数据管理的软件，各程序之间的数据不能相互传递，数据不能重复使用。

文件系统管理阶段 (20 世纪 50 年代后期到 20 世纪 60 年代中后期)，如图 1-10 所示，其特点如下。

- 数据没有完全独立。出现了外存储器 (以下简称“外存”)，操作系统中的文件系统负责对外存上的数据文件进行管理。程序和数据分离，但数据文件专门服务于某一特定程序，仍然不能被其他应用程序共享。
- 存在数据冗余。数据没有合理和规范的结构，使得数据的共享性差。
- 数据不能集中管理。数据文件没有集中的管理机制，数据的安全性和完整性都不能保障。各数据之间缺乏联系，给数据处理带来麻烦。

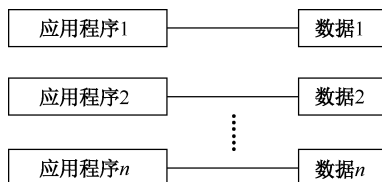


图 1-9 人工管理阶段

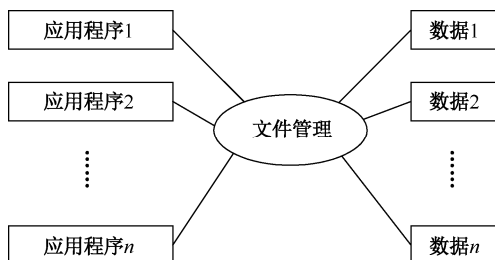


图 1-10 文件系统管理阶段

数据库系统管理阶段 (自 20 世纪 60 年代末至今)，如图 1-11 所示。

为了满足多用户、多程序共享数据的需求，数据库系统将所有的数据集中到一个数据库中实行集中管理。在程序和数据库之间有一个数据库管理系统负责数据库的建立、使用和维护。数据库系统的出现是计算机数据管理技术的重要进步，它的特点如下。

- 支持数据共享，减少数据冗余。
- 数据库中的数据是结构化的 (遵循层次、网状或关系模型)，并由支持该结构的数据库管理系统进行管理和维护。
- 具有较高的数据独立性。数据可以被不同的应用程序共享，应用程序也不必随数据存储结构的改变而变动。
- 具有统一的数据控制功能。数据库管理系统为数据库的使用提供并发访问、安全控制和完整性控制等功能。

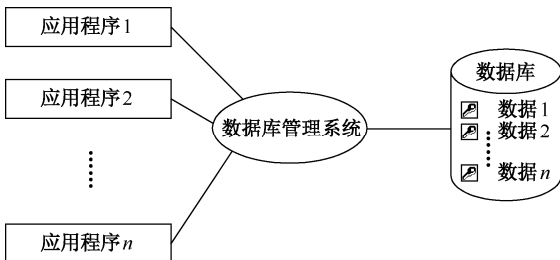


图 1-11 数据库系统管理阶段

2. 数据库设计过程

一个典型的数据库应用系统由数据结构、用户界面、查询和报表等部分组成。在信息系统设计的第一阶段，开发者需要根据用户功能需求明确将来希望从设计的数据库中得到的有用信息，由此可以确定需要使用的相关数据。在数据结构设计的初期阶段，可以使用某些数

据建模技术（如 E-R 模型）来描述需求信息及要存储在数据库中的信息，这一阶段称为数据的概念模型设计阶段。在后面阶段，通过概念模型设计能映射到物理模型上的数据库、数据表及表之间的关系，这一阶段称为逻辑设计阶段。

3. E-R 模型

E-R（Entity-Relationship）模型，即实体-关系模型，是由美籍华裔计算机科学家陈品山于 1976 年首先提出的。它提供不受任何数据库管理系统约束的面向用户的表达方法，在数据库设计中被广泛用作数据建模的工具。

E-R 模型的构成成分包括实体集、属性和联系集 3 部分，其表示方法如下。

实体集用矩形框表示，矩形框内写上实体名，如图 1-12 所示。



图 1-12 E-R 模型中实体表示方法

实体的属性用椭圆框表示，框内写上属性名，并用无向直线与其实体集相连。能够唯一标识实体的属性或属性集可以用实体标识符来表示，实体标识符所代表的属性也就是实体的主键。在 E-R 模型中，添加了下画线的属性名为主键，如学生的学号、课程的编号、读者的借阅证号等，如图 1-13 所示。

实体间的联系用菱形框表示，联系以适当的含义命名，名字写在菱形框中，矩形框分别与菱形框相连，并在连线上标明联系的类型，即 1—1、1—N、M—N。

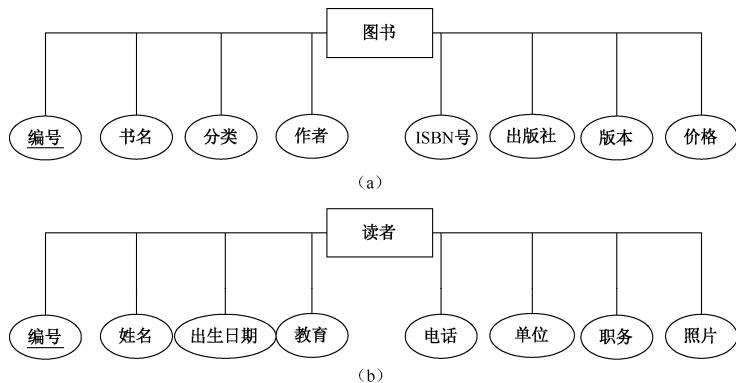


图 1-13 E-R 模型中属性及主键的表示方法

E-R 模型图也称为 E-R 图，如图 1-14 所示。

E-R 图的绘制方法说明：菱形表示实体之间的关系，用矩形表示实体，用无向直线把菱形与有关实体连接，在直线上标明联系的类型。用椭圆表示实体的属性，并用无向直线把实体与属性联系起来。

建立 E-R 模型是数据库概念设计的重要内容，属于软件生命周期的设计阶段，而概念设计是设计阶段的组成部分。

4. 数据模型

数据模型是数据库系统中用于提供信息表示和操作手段的结构形式，是数据库管理系统用来表示实体及实体联系的方法。

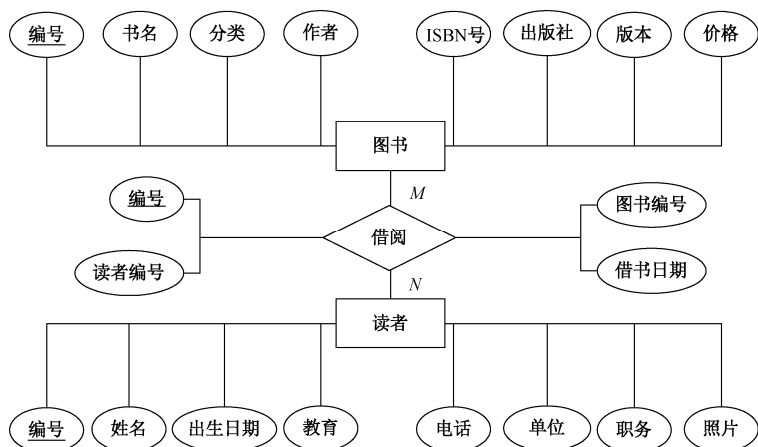


图 1-14 “图书管理信息系统” E-R 模型图

任何一个数据库管理系统都是基于某种数据模型的。常见的数据模型有 3 种：层次模型、网状模型和关系模型。相应的数据库也有 3 种：层次模型数据库、网状模型数据库和关系模型数据库（简称关系数据库）。

(1) 层次模型

层次模型是数据库最早使用的一种数据模型。它是以倒挂的树状结构表示实体及实体间联系的模型。根节点在顶层，子节点在下层，逐层排列，如图 1-15 所示。

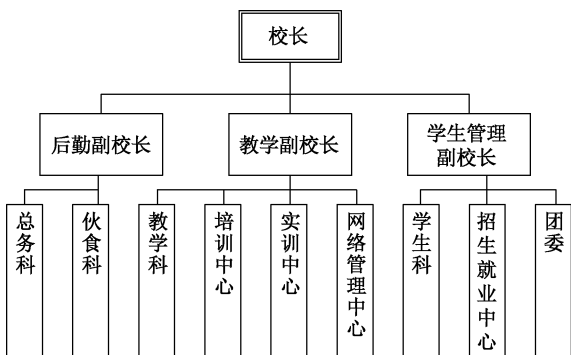


图 1-15 机构设置层次模型示意图

学校中的行政机构、企业中的部门编制等都属于这种结构，它体现了实体间的一对多联系，不能直接表现出多对多关系。

(2) 网状模型

网状模型以网状结构表示实体与实体之间的关系。网状模型可以表示多个从属关系的联系，也可以表示数据间的交叉关系，即数据间的横向关系与纵向关系。网状模型体现了实体间的多对多联系，但数据结构复杂。它是层次模型的扩展。

路由器配置图一般采用网状结构，如图 1-16 所示。

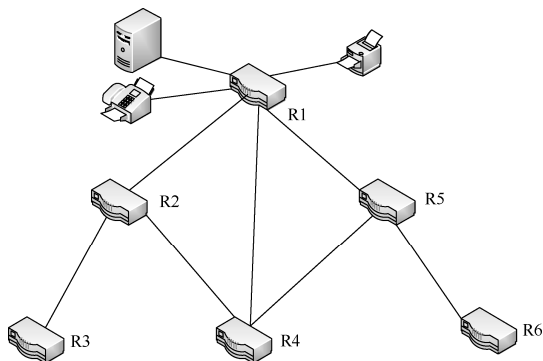


图 1-16 路由器连接网状模型示意图

(3) 关系模型

关系模型以二维表结构来表示实体及实体间的联系。在关系模型中，操作的对象和结果都是二维表，这种二维表就是关系，如图 1-17 所示。

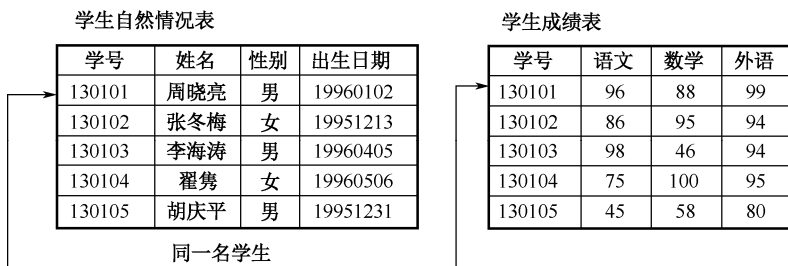


图 1-17 学生实体关系模型示意图

5. 数据库、数据库管理系统、数据库应用系统和数据库系统

数据库 (Data Base, DB): 存储在计算机设备上的结构化的相关数据集合。它不仅包括描述事物的数据本身, 而且还包括相关事物之间的联系。

数据库管理系统 (Data Base Management System, DBMS): 为数据库的建立、使用和维护而配置的软件。该系统能对数据库提供并发控制、安全控制和完整性控制机制及一套能够建立、修改、删除、新增数据库的操作命令和程序开发环境。Visual FoxPro 就是一个可以在计算机和服务器上运行的数据库管理系统。

数据库应用系统 (Data Base Application System, DBAS): 指系统开发人员利用数据库系统资源开发出来的、面向某一实际应用的应用软件系统, 如图书管理信息系统、学生成绩管理信息系统、公司财务报表系统等。

数据库系统 (Data Base System, DBS): 指引进数据库技术后的计算机系统, 提供数据处理和信息资源共享的方式, 能实现有组织、动态地存储大量相关数据的综合系统。它包括 5 部分: 硬件系统、数据库集合、数据库管理系统及相关软件、数据库管理员和用户。

它们之间的包含关系如图 1-18 所示。

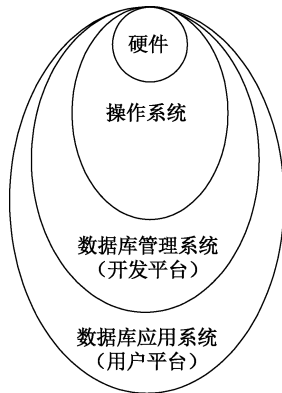


图 1-18 数据库系统层次示意图



项目小结

本项目作为数据库初学者的第一课, 提供了一个数据库模型设计的案例, 引导初学者逐步接触数据库, 了解它的发展过程, 理解数据库及其系统的概念和作用, 掌握数据库设计的方法和原则。本书将要详细介绍的 Visual FoxPro 是在物理上能实现和管理数据库、开发数据库应用系统的数据库管理系统。



知识巩固

一、选择题

1. 在概念模型中, 实体对应了数据库的 ()。



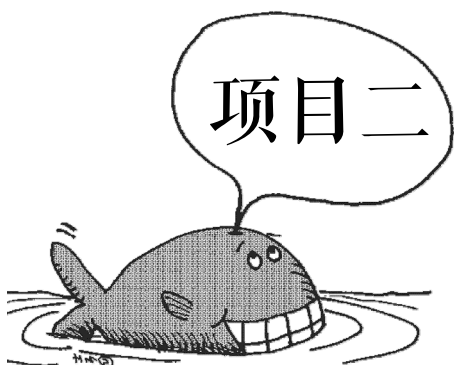
- A. 字段 B. 域 C. 表 D. 记录
2. 概念模型中实体的属性对应了数据表的 ()。
- A. 字段 B. 域 C. 值 D. 记录
3. 在订单管理信息系统中, 商品实体和客户实体之间的关系是 ()。
- A. 一对一 B. 一对多 C. 多对一 D. 多对多
4. 能唯一标识一条记录的字段称为 ()。
- A. 外关键字 B. 主关键字 C. 普通索引字段 D. 索引字段
5. 数据库的设计要考虑尽量避免 () 因素。
- A. 冗余性 B. 完整性 C. 独立性 D. 关联性
6. 域是指属性值的取值范围, 下列对于“年龄”属性的域, 合理的是 (), 单位是“年”。
- A. $-1 < n < 100$ B. $0 < n < 150$ C. $0 \leq n < 150$ D. $20 \leq n < 150$
7. 用数据二维表来表示实体及实体间联系的数据模型称为 ()。
- A. E-R 模型 B. 层次模型
- C. 网状模型 D. 关系模型
8. 数据库、数据库系统、数据库管理系统三者之间的关系是 ()。
- A. 数据库包括数据库和数据库管理系统
- B. 数据库管理系统包括数据库和数据库系统
- C. 数据库包括数据库系统和数据库管理系统
- D. 数据库系统、数据库、数据库管理系统意思相似
9. Visual FoxPro 属于 () 数据库管理系统。
- A. 层次型 B. 网状 C. 关系型 D. 树状
10. 学生选修表中的成绩可以为 $0 \sim 100$ 的实数, 这个范围在关系模型中被称为 ()。
- A. 记录 B. 元组 C. 域 D. 字段

二、填空题

1. 数据库管理技术的发展经历了____、____、____3 个阶段。
2. 实体之间的联系归结为____、____、____3 种。
3. 常用的数据模型有____、____、____3 种类型。
4. 二维表中的列称为关系的____; 二维表中的行称为关系的____。
5. 关系是具有相同性质的____的集合。

三、简答题

1. 如何将实体及实体间的联系设计成关系数据库?
2. 用 E-R 图分别画出订单管理信息系统和综合人事管理信息系统的数据概念模型。



认识数据库管理系统 Visual FoxPro

项目说明

Visual FoxPro 是 Microsoft 公司开发的一个关系型数据库管理系统，可以运行在 Windows XP、Windows Vista、Windows 7、Windows Server 2005 或更高版本操作系统中。Visual FoxPro 系列数据库管理软件的发展演变过程如下。

1980 年，美国 Ashton-Tate 公司推出 dBASE 、dBASE 。

1987 年，Fox 软件公司推出 FoxBASE 1.0、FoxPro 2.0（兼容 dBASE 系统）。

1989 年起，Fox 软件公司推出 FoxPro 1.0、FoxPro 2.0。

1992 年，Microsoft 公司兼并 Fox 软件公司升级推出 FoxPro 2.5 for DOS、Windows。

1995 年，Microsoft 公司推出 Visual FoxPro 3.0。

1997 年，Microsoft 公司推出 Visual FoxPro 5.0。

1998 年，Microsoft 公司推出 Visual FoxPro 6.0，并集成于 Visual Studio 6 软件开发工具包中。

2007 年，Microsoft 公司推出 Visual FoxPro 9.0。

本项目着重介绍 Visual FoxPro 数据库管理软件的使用方法及基本数据类型的操作，分 6 个任务进行实施。

任务一：安装 Visual FoxPro 9.0。

任务二：使用 Visual FoxPro 用户界面。

任务三：在“命令”对话框中执行简单命令。

任务四：创建简单表达式。

任务五：调用常用函数。

任务六：使用项目管理器管理文件。

任务一 安装 Visual FoxPro 9.0

任务目标

能在操作系统下安装 Visual FoxPro 9.0。



能使用多种方式熟练启动和关闭 Visual FoxPro 9.0。

任务描述

操作系统下安装 Visual FoxPro 9.0，并使用多种方式启动和退出该系统。

相关知识

Visual FoxPro 9.0 软件对安装环境的要求如下。

操作系统：Windows XP、Windows Vista、Windows 7、Windows Server 2005 或更高版本的操作系统。

内存：64MB，推荐 128MB 或更高

硬盘：程序安装需要 165MB，额外的 20MB 硬盘空间用于安装系统必备程序。

光驱：CD-ROM 或 DVD-ROM。

显示：VGA800×600 像素，256 色。

鼠标：Microsoft 鼠标或兼容鼠标。



实施步骤

1. 安装

从光驱或网络上选择安装包中的 setup.exe 程序，双击相应图标。

按照提示单击“下一步”按钮，完成安装过程。

2. 启动 Visual FoxPro 9.0

可以使用以下两种方法中的任一种。

选择“开始”“所有程序”命令，选择“Visual FoxPro 9.0”选项。

需要快速启动时，可以在桌面创建 Visual FoxPro 应用程序的快捷方式图标，双击此图标便可以运行。

3. 退出 Visual FoxPro 9.0

可以使用以下 4 种方法中的任一种。

在“命令”对话框中输入“QUIT”，并按 Enter 键。

直接按组合键 Alt+F4。

选择“文件”“退出”命令。

单击标题栏右上角的“关闭”按钮。

注意：正常退出时，Visual FoxPro 会自动保存对数据的修改，或在退出时提示是否保存修改。如果为非正常退出，有可能会丢失数据。

任务二 使用 Visual FoxPro 用户界面

任务目标

熟悉 Visual FoxPro 用户界面。

能够熟练操作 Visual FoxPro 用户界面上的菜单和工具栏。

任务描述

启动 Visual FoxPro 用户界面，能熟练使用鼠标或快捷键操作菜单项，并使用多种方法显示和隐藏工具栏。



实施步骤

1. 启动 Visual FoxPro 9.0

启动 Visual FoxPro 9.0，屏幕显示如图 1-19 所示的集成环境下的用户界面。

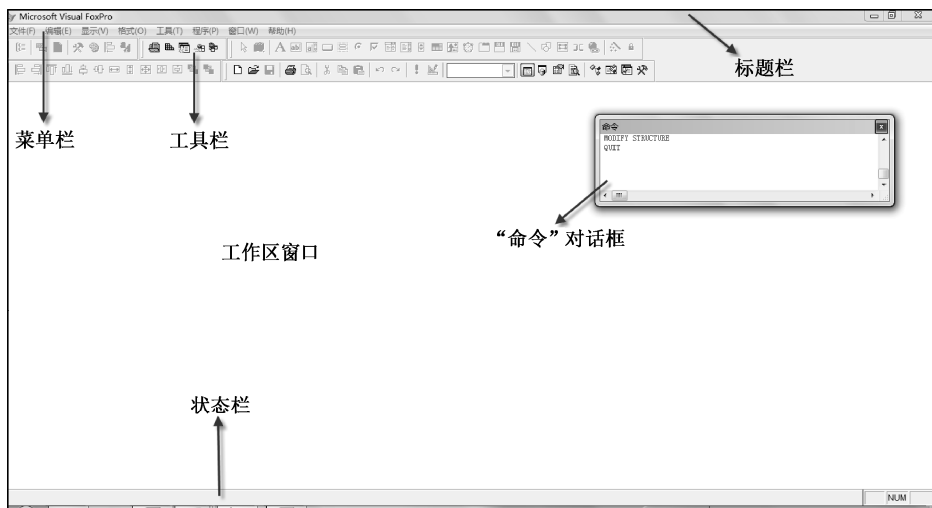


图 1-19 Visual FoxPro 9.0 集成环境下的用户界面

2. 用户界面的组成部分及功能

标题栏：显示软件的名称、图标及窗口控制按钮。

菜单栏：显示 Visual FoxPro 的所有菜单命令。

工具栏：常用菜单命令的按钮显示方式。

工作区窗口：显示各种子窗体的主窗口，程序的运行结果、输出都在这个窗口。

“命令”对话框：Visual FoxPro 用来进行数据库管理的主要方式之一，在该对话框中可以运行各种命令。

3. 操作菜单

(1) 认识菜单项

菜单是包括命令、过程、子菜单的选项，如图 1-20 所示。

(2) 操作菜单项

单键快捷键和组合快捷键。主菜单的快捷键是在菜单项右侧括号里的单个字母，它和 Alt 键组合使用，可以打开菜单项。

菜单项右侧的单键快捷键用于在打开下拉菜单时，通过键盘敲击该字母直接选择菜单项。



菜单项的组合快捷键用于下拉菜单未打开时，通过组合快捷键直接选择菜单项。

带“▶”符号的菜单项是含有子菜单的菜单项。

带省略号的菜单选项表示选择该项后将会弹出一个对话框，等待用户继续选择。

有“ ”符号的菜单选项，表示当前该项有效，单击“ ”符号，可以使它失效。

灰色菜单选项：表示该菜单项当前不能使用。



图 1-20 认识 Visual FoxPro 用户集成环境中的菜单项

4. 操作工具栏

常用的命令通常使用工具栏上的按钮执行。

(1) 显示工具栏

选择“显示” “工具栏”命令，弹出“工具栏”对话框，如图 1-21 所示。

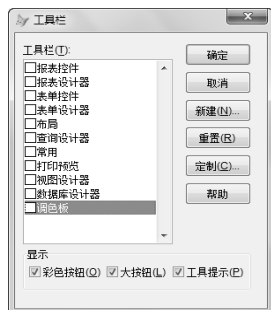


图 1-21 “工具栏”对话框

Visual FoxPro 提供了 11 个工具栏，每个工具栏左侧都有一个复选框，勾选复选框后单击“确定”按钮，即可在窗口中显示该工具栏。Visual FoxPro 用户集成环境中所有工具栏显示的状态如图 1-22 所示。

(2) 隐藏工具栏

单击工具栏右上角的“关闭”按钮。

选择“显示” “工具栏”命令，在“工具栏”对话框中取消勾选复选框。

在工具栏的空白处右击，在弹出的快捷菜单中选择“隐藏”或“显示”命令。

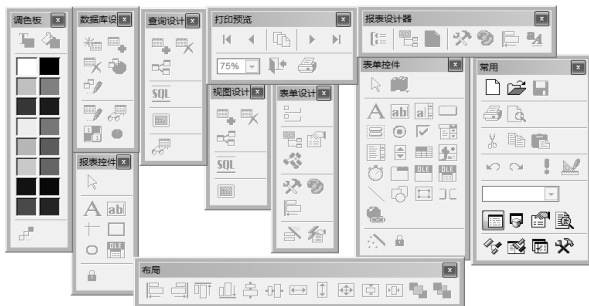


图 1-22 Visual FoxPro 用户集成环境中所有工具栏显示的状态

任务三 在“命令”对话框中执行简单命令

任务目标

- 熟练使用命令操作方式。
- 能够在“命令”对话框中使用输出命令“?”输出简单图形。
- 能够使用命令清空工作区退出 Visual FoxPro 用户界面。
- 能够改变“命令”对话框中文字的格式。

任务描述

使用输出命令“?”输出图形(如图 1-23 所示),然后清除,退出 Visual FoxPro 系统。

相关知识

1. Visual FoxPro 的工作方式

- 在“命令”对话框中选择命令。
- 通过菜单或工具栏选择命令。
- 执行应用程序。

2. “命令”对话框的打开和关闭方法

一般在启动 Visual FoxPro 软件后,“命令”对话框的默认状态是打开的。也可以通过选择“窗口”“命令窗口”命令弹出“命令”对话框。通过单击“命令”对话框右上角的“关闭”按钮可以关闭“命令”对话框。

3. 命令的分类

常用 Visual FoxPro 命令按照功能不同,大致可以分为五大类:数据库操作命令、输入/输出命令、程序设计命令、共享访问数据命令和 Visual FoxPro 环境命令。举例说明如下。

(1) 数据库操作命令

DISPLAY DATABASE

```

*
***
*****
*****
*****
*
*
*****
*****
*****
*****
*

```

图 1-23 图形



DELETE DATABASE

&&显示有关当前数据库信息

&&从磁盘上删除数据库

(2) 输入/输出命令

? &&计算表达式并将结果发送到该 Visual
DISPLAYFoxPro 主窗口、活动的自定义窗口或打印机
&&显示当前表的记录内容和表达式结果

(3) 程序设计命令

DO

&&执行程序

RETURN

&&返回命令

(4) 共享访问数据命令

SET EXCLUSIVE

&&指定在网络上以独占方式还是共享方式打开表文件

UNLOCK

&&释放一个表或所有打开表中的单记录锁、多记录锁、文件锁

(5) Visual FoxPro 环境命令

DIR

&&显示目录中有关文件的信息

HELP

&&打开“帮助”窗口

QUIT

&&结束当前 Visual FoxPro 工作期, 并将控制权返回给操作系统

4. 命令的格式

Visual FoxPro 命令通常由两部分组成, 前一部分是命令动词, 表示应该执行的操作, 后一部分是若干短语, 对操作动作提供某些限定条件。其一般格式如下。

命令动词 [范围] [表达式] [FIELDS <字段名 1, 字段名 2.....>] [FOR/WHILE<条件>]

命令在前, 短语在后。短语中各子句可以互换位置, 以空格相隔。命令动词和短语中的保留字可以简写为前 4 个字符, 且不区分大小写。

符号的含义: []——可选项; < >——必选项; /——二选一。

范围: 对记录的选择, 常用表示范围的保留字如下。

ALL: 数据表中所有记录。

NEXT <N>: 包括当前记录在内的后 N 条记录。

RECORD <N>: 记录号为 N 的一个记录。

REST: 从当前到最后的记录。

条件。FOR<条件表达式>: 使一条命令作用于满足指定逻辑条件的每条记录。WHILE <条件表达式>: 使命令作用于只要该逻辑表达式计算为真(T.)的每条记录。一旦该表达式计算为假(F.), 该命令即停止运行, 不对剩余的记录进行操作。

5. 命令的编辑

命令行中的命令一定是半角状态下输入的字符, 否则系统会提示错误, 如图 1-24 所示。命令很长时, 可以分多行书写, 在命令断开处输入分号, 然后按 Enter 键来分断很长的命令。

在“命令”对话框中, 可以通过选择“格式”菜单中适当的命令来改变字体、行间距及缩进。

在“命令”对话框是编辑对话框, 可以在“命令”对话框中编辑、插入、删除、剪切、复制或粘贴文本, 如图 1-25 所示。

6. 命令的执行

命令的执行有以下几种方法。

在“命令”对话框中直接输入命令，然后按回车键即可执行。

在“命令”对话框中运行同一代码行，可以将光标移到要运行代码行中的任何位置，并按 Enter 键执行。

要在“命令”对话框中运行多个代码行，选择要运行的多行代码，按 Enter 键。

选择菜单项中的命令时，系统会自动在“命令”对话框中显示命令。



图 1-24 全角字符命令在执行时提示错误信息

图 1-25 “命令”对话框中的编辑菜单



实施步骤

依次在“命令”对话框中输入如下命令。

```
? "      *      " (按 Enter 键)
? "      ***      "
? "      *****      "
? "      *********      "
? "      *****      "
? "      ***      "
? "      *      "
```

选中上一步分步运行的七行代码，按 Enter 键运行代码块。

在“命令”对话框中输入以下命令。

```
CLEAR (按 Enter 键)
```

在“命令”对话框中输入以下命令。

```
QUIT (按 Enter 键)
```

清空“命令”对话框中的所有命令。

将光标移动到“命令”对话框中并右击，在弹出的快捷菜单中选择“清除”命令，如图 1-26 所示。



问题探究

1. 使用菜单或工具栏选择命令和在“命令”对话框中使用命令有什么不同？

单击某个菜单项或工具栏的某个按钮时，系统自动调用并选择相应命令。菜单和工具栏所能选择的命令都是较为常用的命令。Visual FoxPro 提供了数百条命令，很多命令在菜单和工具栏中找不到，需要在“命令”对话框中使用。

2. 如何将本任务中的输出命令以应用程序的方式执行？

Visual FoxPro 的应用程序是以“.prg”为扩展名的文件。在“命令”对话框中通过输入命令“MODIFY COMMAND 文件名.prg”可以创建并打开一个命令文件，在该文件的编辑窗口



图 1-26 选择“清除”命令



输入本任务中的所有命令后保存关闭。然后在“命令”对话框中选择“DO 文件名.prg”命令，即可输出本任务中的运行结果。

任务四 创建简单表达式

任务目标

能够正确辨别并书写字符型、数值型、日期型、日期时间型、货币型、逻辑型的常量。

能够正确定义内存变量、数组并赋值。

能够根据情境熟练使用算术运算符、字符运算符、日期时间运算符、关系运算符和逻辑运算符组成表达式。

任务描述

创建简单表达式，完成如下任务，并在“命令”对话框中执行。

假设学生 A 的计算机网络课程成绩为 90，用字母 AC 表示；学生 B 的成绩为 98，用字母 BC 表示，如何判断 AC 和 BC 是否相等？

假设学生 A 的计算机网络课程成绩为 90，用字母 AC1 表示；Java 程序设计课程成绩为 87，用字母 AC2 表示；网页设计课程成绩为 97，用字母 AC3 表示，如何表示三科的总成绩？

假设学生 A 的出生年月是 1992 年 4 月，如何表示学生 A 的年龄？

学校对班级名称的命名方法是专业名称和入学年份后两位的组合，如何表示 2014 年入学的计算机应用专业的班级名称？

假设学生 A 是女生，学生 B 不是女生，“学生 A 和学生 B 都是女生”这句论断正确吗？如何表示？

相关知识

1. 数据类型

Visual FoxPro 的数据包括常量、变量、数组、字段、记录、对象等，并且对这些数据提供了 6 种常用数据类型，它们是字符型（C）、数值型（N）、日期型（D）、货币型（Y）、日期时间型（T）、逻辑型（L），不同数据类型的数据书写格式不同，可以进行的运算也不同。

2. 常量

常量是值始终保持不变的值，常量的类型和格式如表 1-1 所示。

表 1-1 常量的类型和格式

常量类型	格式	字节数	组成	举例
字符型	" "、' '、[]		数字、字母、空格、汉字等	" 32 "、" 汉 "、[计算机 96]
数值型	整数、实数	8	正负号、数值、小数点、乘幂字母 E 或 e	-51、20.6E12、53

续表

常量类型	格式	字节数	组成	举 例
货币型	“\$”符号引导的数值，四位小数位	8	字符“\$”、数值、正负号	\$123.5210
日期型	系统默认格式{^yyyy-mm-dd}或者{^yyyy/mm/dd}，可以自行设置	8	数值、格式符{^}、分隔符	{^1984/05/12}、{^09-30-13}
日期时间型	系统默认格式{^yyyy-mm-dd [hh[:mm[:ss]]] [am pm]}	4	数值、格式符{^}、分隔符、固定字符(“am”代表上午，“pm”代表下午)组成	{^1998-09-21 12:34:10 am}
逻辑型	.T.、.t.、.Y.、.y.或者 .F.、.f.、.N.、.n.		句点(.) 字符“T”、“Y”、“F”、“N”，不区分大小写	

3. 变量

变量是在程序运行过程中值可以发生改变的量。

变量分为字段变量和内存变量两大类。字段变量随着表的存取而存取，是永久变量。内存变量是临时变量，随着程序的运行而加载到内存，程序运行完毕后被系统从内存中自动释放。它包括简单变量、数组变量和系统变量。

(1) 简单变量：只能存储一个数据的变量。简单变量存储的数据可以是任何一种数据类型。

变量的命名规则。变量名可以使用字母、汉字、下画线和数字，开头字母必须是字母、下画线或汉字，中间不能有空格。变量名应尽量不与保留字、字段变量、系统变量重复。

变量的赋值。简单变量可以不用申明就赋值，可以多次给一个变量赋予多个数据类型，变量类型就是它所存储的数据的类型。给变量赋值的方法有两种，一种命令格式如下。

```
STORE <表达式> TO <内存变量表>
```

功能：计算表达式的值存储到内存变量中，内存变量表中的变量用逗号相隔。例如：

```
STORE "234" TO MyChar
```

另一种命令格式如下。

```
<内存变量表>=<表达式>
```

功能：计算表达式的值并赋值给内存变量表中的变量，功能等同于 STORE 命令。例如：

```
MyChar="234"
```

变量的显示，格式如下。

```
DISPLAY|LIST MEMORY [LIKE <通配符>][TO PRINTER|TO FILE<文件名>]
```

功能：显示内存变量的当前信息，包括变量名、作用域、类型、取值信息。

参数：

- DISPLAY 是分屏显示，LIST 是连续显示。
- LIKE <通配符>：通配符可以是星号(*)，也可以是问号(?)。星号(*)代表0到多个字符，问号(?)代表一个字符。
- TO PRINTER：输出结果到打印机。
- TO FILE<文件名>：输出结果到指定的文件，并保存。

【例 1-5】

```
STORE "We are Chinese" TO A,AB,AAC
AB1=$125.3221
DISP MEMO
LIST MEMO TO D:\MEMO.TXT
DISP MEMO LIKE A
```

运行结果如图 1-27 所示。



图 1-27 显示内存变量执行结果

变量的删除。

命令格式：CLEAR MEMORY

功能：删除当前内存中所有内存变量。

命令格式：RELEASE <内存变量表>

功能：清除内存变量表中的变量，释放相应内存空间。

【例 1-6】

```
A=100
B=120
DISP MEMO          &&结果显示中包含内存变量 A、B
RELEASE B
DISP MEMO          &&结果显示中没有内存变量 B
```

(2) 数组变量

数组的声明。数组是用来存储一组相关数据的结构化的变量。这些变量具有相同的变量名，通过下标来区分不同的变量。Visual FoxPro 只允许定义一维和二维数组，且数组在使用前必须经过声明。数组声明的命令格式如下。

DIMENSION|DECLARE<数组名 1>(行标上界表达式[, 列标上界表达式])[, 数组名 2

功能：声明一个或多个数组。

说明：

- 数组下标的下界为 1。
- 数组定义后每个数组变量可以单独使用。
- 同一数组中各变量的数据类型可以不同。
- 二维数组按照行进行存储，可以按照一维数组变量形式使用。例如，A(3)(4)可以用 A(12)代替。
- 下标可以使用中括号[]或者圆括号()将数字括起来。

数组变量的使用。

【例 1-7】

```
DIMENSION SAMPLE(2,3)
STORE 'Goodbye' TO SAMPLE(1,2)
STORE 'Hello' TO SAMPLE(2,2)
STORE 99 TO SAMPLE(6)
```



```
STORE .T. TO SAMPLE(1)
CLEAR
DISPLAY MEMORY LIKE SAMPLE
```

运行结果如图 1-28 所示。

(3) 系统变量

系统变量是由 Visual FoxPro 系统自动生成和维护的内存变量。它由下画线 “_” 打头，每一个系统变量用来储存一个系统的状态或特性。例如，系统变量 _ALIGNMENT 表示对齐方式，_DIARYDATE 保存当前系统日期。

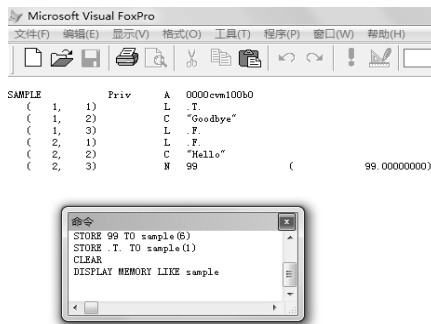


图 1-28 程序运行结果

4. 运算符与表达式

运算符是对数据进行加工处理的符号。例如，常用的算术运算符有+、-、×、/。

表达式是通过运算符将各种不同类型的数据（包括常量、变量、字段、函数等）连接起来的运算式。运算符的两侧必须是相同类型的数据，表达式的运算结果是具有某个数据类型的值。

常用的运算符分为算术运算符、字符运算符、日期时间运算符、关系运算符和逻辑运算符 5 种，通过这 5 种运算符分别可以形成算术表达式、字符表达式、日期时间表达式、关系表达式和逻辑表达式。在 Visual FoxPro 系统中，各类运算的优先级按从高到低的排序是圆括号、算术和日期运算、字符运算、关系运算、逻辑运算。

(1) 算术运算符和算术表达式

算术运算符按照优先级从高到低的顺序运算，如表 1-2 所示。算术表达式是算术运算符连接数值型数据形成的运算式，运算结果仍然为数值型数据。

表 1-2 算术运算符的优先级

优 先 级	运 算 符	说 明
1	() -	括号、负号
2	**或^	乘方
3	*/、%	乘、除、取余
4	+、-	加号、减号

【例 1-8】

在“命令”对话框中输入如下命令，并求输出结果。

```
A=10
B=2
?(5*A+-10) % (2^B)
```

运行结果：

0

提示：取余运算符“%”的运算规律是被除数的小数位数是结果的小数位数，除数的运算符号是结果的运算符号。例如：6%-4=-2，-2.4%-2=-0.4，-2.4%2=1.6，-2.4%-2=-0.4。

(2) 字符运算符和字符表达式

字符运算符用于连接两个字符型数据进行运算。常用的字符运算符及用法如表 1-3 所示。

字符表达式是由字符运算符连接字符型数据对象（包括字符型常量、变量、函数、字段）



形成的运算式，运算结果为字符型或逻辑型常量。

表 1-3 常用的字符运算符及用法

运 算 符	功 能	结果数据类型
+	保留字符串中的空格位置连接两个字符串	字符串
-	将第一个字符串末尾的空格放在第二个字符串的末尾，再与第二个字符串连接	字符串
\$	比较两个字符串的包含关系，如果第一个字符串是第二个字符串的子串，则返回逻辑值“真”，否则为“假”	逻辑型

【例 1-9】在“命令”对话框中输入如下命令，并求输出结果。

```
? "您" + "好 !"
? "您" - "好 !"
? "您" - "好 "
? "计算机"$ " 计算机专业"
? "计算机"$ "计算机专业 "
? "计算机"$ "计 算 机"
? "计算机"$ "应用计算机"
```

运行结果如图 1-29 所示。

（3）日期时间运算符和日期时间表达式

日期时间运算符是用来连接日期型、日期时间型或数值型数据的操作符，它只有“+”和“-”两种，由此形成的表达式的格式有限定，不能任意组合。表达式的运算结果是日期时间型或数值型。日期时间表达式的格式及功能如表 1-4 所示。

表 1-4 日期时间表达式的格式及功能

格 式	功 能	结果数据类型
日期+天数	以“日期”为起始，计算“天数”后的日期	日期型
天数+日期		
日期-天数	以“日期”为起始，计算“天数”前的日期	日期型
日期-日期	两个日期之间相差的天数	数值型
日期时间+秒数	以“日期时间”为起始，计算“秒数”后的日期时间	日期时间型
秒数+日期时间		
日期时间-秒数	以“日期时间”为起始，计算“秒数”前的日期时间	日期时间型
日期时间-日期时间	两个日期时间之间相差的秒数	数值型

【例 1-10】在“命令”对话框中输入如下命令，并求输出结果。运行结果如图 1-30 所示。



图 1-29 字符表达式运行结果

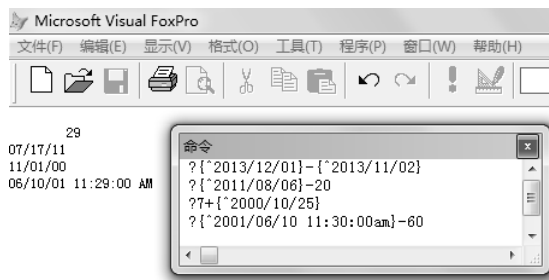


图 1-30 日期时间表达式运行结果

```
? {^2013/12/01}-{^2013/11/02}
? {^2011/08/06}-20
? 7+{^2000/10/25}
? {^2001/06/10 11:30:00am}-60
```

运行结果：

```
oT/17/11/11/01/00/60/10/01 11.29:00AM
```

(4) 关系运算符和关系表达式

关系运算符是用来完成比较运算的运算符。可以将两个算术表达式、字符表达式、日期表达式、逻辑表达式进行比较，运算结果为逻辑型常量“真”或“假”。常用的关系运算符如表 1-5 所示。

表 1-5 常用的关系运算符

运 算 符	说 明	结果数据类型
<	小于	逻辑型常量
>	大于	
=	等于，当比较字符串时，当等号右边的字符串与左边的字符串的前部分相匹配，则比较结果为真，否则为假	
<>	不等于	
#		
!=		
<=	小于等于	
>=	大于等于	
==	字符串精确比较，只有在等号左右的字符串完全相等时，结果才为真，否则为假	

关系运算符比较的基本原则如下。

数值型数据和货币型数据的比较按照数值的大小进行。例如， $20 < -2$ 的结果为假， $\$10.3200 > \10.3000 的结果为真。

日期型、日期时间型按照年、月、日的前后顺序比较，时间早的值小。例如， $\{^1996/06/05\} > \{^2000/01/01\}$ 的结果为假。

逻辑值中，.T.大于.F。

单个字符的比较是按照字符的 ASCII 码的大小进行比较的。字符串是按照从左往右的顺序逐字符进行比较，直到遇到不相同的字符，由该字符确定大小。对于汉字的比较，系统默认按照汉语拼音的顺序进行比较。例如，“A”<“a”的结果为真，“A”<“9”的结果为假，“xyzA”=“xyza”的结果为假，“xyzA”=“xyz”的结果为真，“xyzA”=“xyzL”的结果为假。

(5) 逻辑运算符和逻辑表达式

逻辑表达式是包含逻辑运算符和逻辑型数据的运算式。逻辑运算符包含与、或、非，如表 1-6 所示。

表 1-6 逻辑运算符

运 算 符	说 明	结果数据类型
NOT、.NOT、!	逻辑非	NOT .T.=.F. NOT .F.=.T.
AND、.AND.	逻辑与	.T. AND .T.=.T. .T. AND .F.=.F. .F. AND .T.=.F. .F. AND .F.=.F.



续表

运 算 符	说 明	结果数据类型
OR、.OR.	逻辑或	.T. , OR .T.=.T. .T. , OR .F.=.T. .F. , OR .T.=.T. .F. , OR .F.=.F.

任务分析

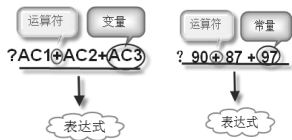


图 1-31 表达式的组成

表达式通过计算后可以是任何一种数据类型的值,通过使用输出命令“?”显示表达式的值来进行测试。

以任务描述中第 2 个问题为例进行分析, AC1、AC2、AC3 是 3 个内存变量,分别存储学生 A 的三门课程成绩,通过使用算术运算符“+”进行数值求和运算(如图 1-31 所示),求三门课程总成绩,并通过“?”命令输出运行结果。



实施步骤

在“命令”对话框中输入下列命令,并输出结果,如图 1-32 所示。

在“命令”对话框中输入下列命令,并输出结果,如图 1-33 所示。

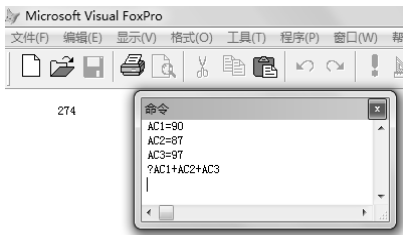


图 1-32 运行结果一

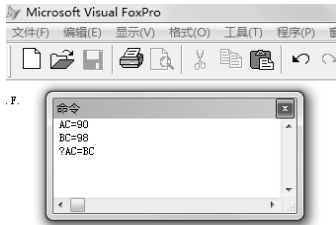


图 1-33 运算结果二

在“命令”对话框中输入下列命令,并输出结果,如图 1-34 所示。

在“命令”对话框中输入下列命令,并输出结果,如图 1-35 所示。

在“命令”对话框中输入下列命令,并输出结果,如图 1-36 所示。

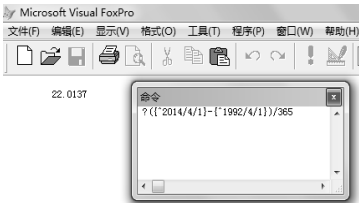


图 1-34 运行结果三



图 1-35 运行结果四

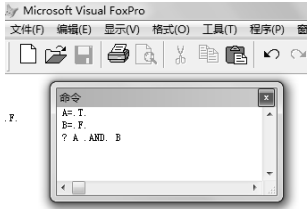


图 1-36 运行结果五



问题探究

1. 表达式可以作为一条语句单独运行吗?

表达式不可以作为一条语句执行。表达式通过运算后是有返回值的,它可以作为赋值语

句的一部分，也可以作为另一个表达式的一部分，它的结果可以通过一个命令输出，语句格式如下。

命令名 <表达式>

2. Visual FoxPro 中有一组 SET 命令，哪些命令对日期型常量的显示有作用？

用户使用日期型常量的系统默认格式是 {^YYYY-MM-DD} (其中分隔符可以为反斜杠 "/" 或点号 ".")。

- SET DATE [YMD|MDY|DMY|.....]：设置日期显示格式为“年月日”或“月日年”或“日月年”。

- SET CENTURY ON|OFF：设置日期中年份的显示格式是 4 位或者两位。

- SET MARK TO<字符串表达式>：可以根据用户需要设置显示日期中年月日数值间的分隔符。

3. SET EXACT ON|OFF 命令对关系运算符 “=” 和 “==” 的作用是什么？

SET EXACT ON：先在较短字符串的尾部添加空格使两个字符串长度相等，再进行精确比较。

SET EXACT OFF：只要右边字符串与左边字符串的前面部分内容匹配，即为真。

这两个命令只对 “=” 有限定作用。

“==” 精确比较：只有当两个字符完全相等时才为真。

【例 1-11】

```
SET EXACT OFF
STORE "计算机" TO S1
STORE "计算机 " TO S2
STORE "计算机世界" TO S3
?S1=S3,S3=S1,S1=S2,S2=S1,S2==S1
```

运行结果：

```
.F. .T. .F. .T. .F.
SET EXACT ON
?S1=S3,S3=S1,S1=S2,S2=S1,S2==S1
```

运行结果：

```
.F. .F. .T. .T. .F.
```

任务五 调用常用函数

任务目标

能够熟练使用常用数值函数、字符函数、测试函数、日期时间函数和数据转换函数解决问题。

任务描述

数值函数的应用：在英语考试成绩中，学生甲的成绩为 98 分，学生乙的成绩为 67 分，学生丙的成绩为 59.5 分，3 个学生中英语最低分是多少？最高分是多少？是否全部及格了？使用常用函数列出表达式并输出结果。

字符函数的应用：已知学生成绩文件的存放路径是 “D:\cjkcj.dbf”，将该路径字符去掉前导和尾部空格后全部转换为大写字母，再从中取出文件名字符串的表达式并输出结果。



日期时间函数的应用：输出报表名称，格式是“××××年 13 级计算机应用专业成绩统计表”，结尾处标明打印时间，格式为“××××年××月××日”。

数据类型转换函数的应用：同学 A 的身高为 1.76m，同学 B 的身高为 1.80m，同学 C 的身高为 1.65m，且保存的数据格式为字符型，请输出他们的平均身高。

测试函数的应用：判断姓名变量 NAME 的值若为空，则返回常量“JACK”，否则返回“JACK”和 NAME 的连接值。

相关知识

1. 函数的定义及调用

函数是预先定义好的能够完成特定功能且具有返回值的一段程序。函数可以分为系统函数和自定义函数。系统函数是 Visual FoxPro 数据库管理系统提供的函数，系统函数库包括数百种函数供用户调用。自定义函数是用户根据需求自己定义的函数。自定义的函数名不能与系统函数名同名，且必须先定义后使用。

函数定义的格式：

```
FUNCTION <函数名> ( [<参数表>] )  
.....  
RETURN Value  
ENDFUNC
```

函数调用的格式：

函数名 ([<参数表>])

参数表中的参数用逗号相隔，若没有参数，小括号不能省略。函数执行后会返回一个具有固定数据类型的值，称为函数值。函数可以作为表达式运算的对象，也可以作为命令使用。

2. 系统函数的分类

根据函数实现的功能和数据处理类型的不同，将常用系统函数分为数值函数、字符函数、测试函数、日期时间函数和数据转换函数。常用函数的介绍详见本项目阅读材料。

任务分析

常用数值函数 Max (<表达式表>) Min (<表达式表>) 分别返回表达式中的最大、最小值。再判断成绩中的最小值是否小于 60，确定 3 个学生成绩是否全部及格。

经分析，解决该问题需要用到如下常用字符函数。

AllTrim (<字符串表达式>)：执行后删除字符串参数的前导和后缀空格并返回。

Upper (<字符串表达式>)：将字符串表达式中的小写字母全部转换成大写字母后返回，其他字符不变。

RAT (<字符表达式 1> , <字符表达式 2> [, <数值表达式>])：返回字符表达式 1 在字符表达式 2 内最后 (最右边) 出现的位置。假设其中 N 为数值表达式，当 N 大于 1 时，返回从右边查找第 N 次时的位置。

SubStr (<字符串表达式> , <数值表达式 1> , [<数值表达式 2>])：从字符串表达式中第 N 位开始，取 M 个字符作为子串返回 (其中 N 为数值表达式 1 , M 为数值表达式 2)。当 <数值表达式 2> 省略时，返回的子串从第 N 位开始到最后一位。

Date()：返回当前计算机系统的日期时间。函数 Year (<日期型表达式>) 返回日期型

参数中的年份, Month (<日期型表达式>) 返回日期型参数中的月份、Day (<日期型表达式>) 返回日期型参数中的日期。通过将 Date() 函数返回的当前时间作为函数 Year()、Month()、Day() 的参数分别得到打印报表中的数值。

数据保存类型为字符型, 要计算平均值, 必须先进行类型转换。需要用到的函数如下。

Val (<字符串表达式>): 将字符串表达式中由数字、正负号及小数点组成的字符串转换成数值型数据并返回。

经分析, 解决该方法可以使用条件测试函数。

IIF (<逻辑表达式>, <表达式 1>, <表达式 2>): 测试逻辑表达式的值, 若为真, 返回表达式 1 的值; 否则, 返回表达式 2 的值。



实施步骤

在“命令”对话框中输入如下命令, 并返回运行结果。

```
?Min (98,67,59.5)
?Max (98,67,59.5)
?Min (98,67,59.5)>=60
```

运行结果:

```
59.5 98 .F.
```

在“命令”对话框中输入如下命令, 并返回运行结果。

```
MyStr=AllTrim("D:\cjk\Cj.dbf")
UppMyStr=Upper(MyStr)
i=RAT('\',UppMyStr)
?SubStr(UppMyStr,i+1)
```

运行结果:

```
CJ.DBF
```

在“命令”对话框中输入如下命令, 并返回运行结果。

```
NowDate=Date()
NowYear=Year(NowDate)
NowMonth=Month(NowDate)
NowDay=Day(NowDate)
ReportName=Str(NowYear)+"年13级计算机应用专业成绩统计表"
ReportEnd=Str(NowYear,4)+"年"+Str(NowMonth,2)+"月"+Str(NowDay,2)+"日"
?ReportName
?ReportEnd
```

运行结果:

```
2013 年 13 级计算机应用专业成绩统计表
2013 年 12 月 26 日
```

在“命令”对话框中输入如下命令, 并返回运行结果。

```
A="1.76"
B="1.80"
C="1.65"
?(Val(A)+Val(B)+Val(C))/3
```

运行结果:

```
1.7367
```

在“命令”对话框中输入如下命令, 并返回运行结果。

```
NAME=" "
?IIF(Empty(NAME),"JACK","JACK "+NAME)
```

运行结果:



任务六 使用项目管理器管理文件

任务目标

掌握项目管理器的功能，并能够使用项目管理器创建项目，编译并运行应用程序。

任务描述

创建学生成绩管理信息系统项目文件 Student.pjx，并在项目管理器中新建主程序文件 Main.prg 和 Draw.prg，主程序文件 Main.prg 调用 Draw.prg，连编该项目文件生成可执行文件 Student.exe，运行该程序。

相关知识

项目是相关文件、程序、数据的一个集合。项目管理器是 Visual FoxPro 提供的一个可视化项目管理工具。它能够有效组织、新建、添加、删除、编辑和运行项目中的文档和数据。



图 1-37 项目管理器

项目管理器提供了对同一个项目不同数据的分层结构组织形式。它包含 6 个标签：“全部”、“数据”、“文档”、“类”、“代码”和“其他”，如图 1-37 所示。“全部”标签里按照文件夹形式管理了所有项目数据。“数据”标签包含项目中的数据库、自由表和查询。“文档”标签包含项目中的表单、报表和标签。“类”标签包括项目中的自定义类库。“代码”标签包括程序、API 库和应用程序。“其他”标签包括菜单、文本文件和其他文件。项目保存后形成扩展名为“.pjx”和“.pjw”的文件类型。



实施步骤

1. 创建项目文件

创建学生成绩管理信息系统项目文件 Student.pjx。

1) 执行下列任何一种操作方式，打开“新建”对话框。

单击“常用”工具栏中的“新建”按钮。

选择“文件”→“新建”命令。

按 Ctrl+N 组合键。

2) 在“新建”对话框中选中“项目”单选按钮，单击“新建文件”按钮如图 1-38 所示。

3) 在弹出的“创建”对话框中选择文件保存位置，并输入文件名“student.pjx”，单击“保存”按钮，如图 1-39 所示。



图 1-38 选中“项目”单选按钮

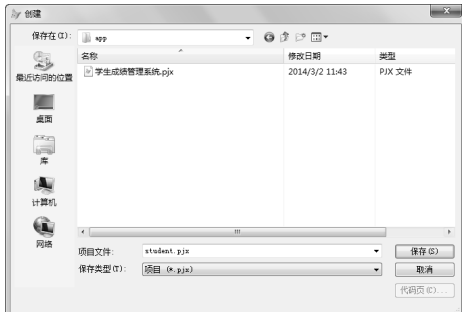


图 1-39 “创建”对话框

2. 新建程序文件 draw.prg

在项目管理器中选择“代码”标签，选择“程序”选项，单击“新建”按钮，打开程序编辑窗口，输入如图 1-40 所示的命令，并保存为 draw.prg 文件。

3. 新建主程序文件 main.prg，并调用 draw.prg

在项目管理器中选择“代码”标签，选择“程序”选项，单击“新建”按钮，打开程序编辑窗口，输入如图 1-41 所示的命令，并保存为 main.prg 文件。

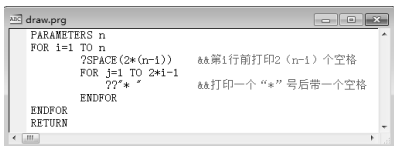


图 1-40 在项目管理器中新建程序文件

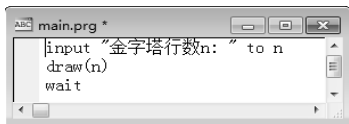


图 1-41 在项目管理器中新建主程序

选择“程序”选项下的 Main 文件，选择“项目”，“设置主文件”命令，如图 1-42 所示。

4. 将项目连编成可执行文件 student.exe

在项目管理器中单击“连编”按钮，弹出“连编选项”对话框，选中“win32 可执行程序/COM 服务程序 (exe)”单选按钮，并勾选“重新编译全部文件”复选框，单击“确定”按钮 (如图 1-43 所示)，将生成 student.exe 文件。

5. 运行文件

运行可执行文件，运行结果如图 1-44 所示。

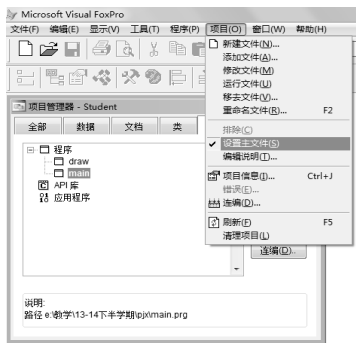


图 1-42 将 main.prg 设置为主文件



图 1-43 将项目所包含的所有程序和文件连编成可执行文件

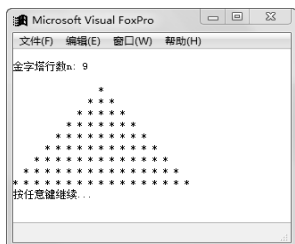


图 1-44 程序运行结果



问题探究

1. 为什么有时运行会提示“文件不存在”的错误？如何解决？

这是因为主程序 Main 在调用 draw.prg 时首先会在系统默认路径下寻找该程序,如果没有,则在用户添加的查找路径中寻找,如果还是没有找到,则会提示如图 1-45 所示错误。解决方法是,选择“工具”→“环境管理器”命令,在弹出的“环境管理器”对话框中默认目录中重新指定或新增一条搜索路径,该路径指向程序 draw.prg,如图 1-46 所示。

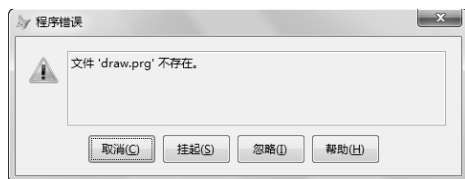


图 1-45 “程序错误”对话框

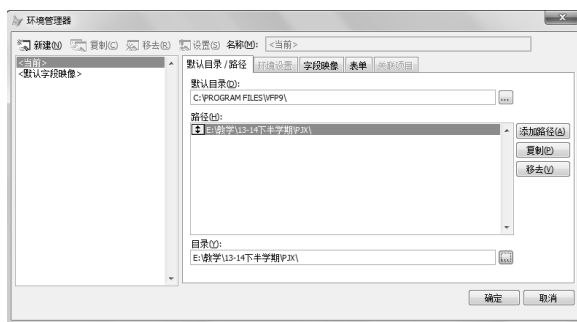


图 1-46 在“环境管理器”对话框中设置默认路径

2. 项目主文件可以有几个？它的作用是什么？

在一个项目中只能有一个主文件,它是该项目运行时的入口文件,可以是程序文件或表单文件。一般在主文件中进行初始化环境设置和配置、清除内存的工作,并通过调用菜单、表单等来显示用户界面。项目管理器处于打开状态时,主菜单会出现“项目”菜单项,在其下拉菜单中,可以设置主文件。

3. 如何通过项目管理器创建和修改各种类型的文件？

项目管理器包含了 6 个标签,分别对数据库、数据表、查询、视图、表单、报表、标签、菜单、程序、文本、类库等文件进行分组管理。

若是第一次创建这些类型的文件,在项目管理器中选择待创建的文件类型,再单击“新建”按钮,即可以启动 Visual FoxPro 提供的相应向导。通过创建向导,系统引导用户分步骤完成创建文件的任务。Visual FoxPro 数据库系统提供的主要向导有项目向导、数据库向导、表向导、查询向导、视图向导、表单向导、报表向导、程序向导、菜单向导等。

若对已有文件进行修改,在项目管理器中选择待修改的文件,再单击“修改”按钮,系统将打开相应的设计器。设计器是系统提供的用来创建和修改各种类型文件的可视化操作工具。Visual FoxPro 数据库系统提供的主要设计器有数据库设计器、查询设计器、视图设计器、表单设计器和报表设计器。



阅读材料

Visual FoxPro 数据库管理系统提供的常用函数可以分为 5 类。

1. 数值函数

(1) 整数函数

函数格式：

Int(<数值表达式>)

功能：返回<数值表达式>的整数部分。

【例 1-12】

?Int(4.6), Int(-4.6)

运行结果：

4 -4

(2) 四舍五入函数

函数格式：

Round(<数值表达式 1>, <数值表达式 2>)

功能：返回数值表达式 1 按照数值表达式 2 的值进行四舍五入后的值。具体方法：当数值表达式 2 为非负数时，对数值表达式 1 小数点后第 N 位四舍五入；否则，对数值表达式 1 小数点前第 N 位四舍五入（ N =数值表达式 2+1）。

【例 1-13】

?Round(2324.8712,2), Round(2324.8712,-2)

运行结果：

2324.87 2300

(3) 最大值函数

函数格式：

Max(<表达式表>)

功能：表达式表由相同数据类型的表达式组成，用逗号相隔。返回表达式表中的最大值。若数据类型是日期时间型，则返回最后面的日期或时间。

【例 1-14】

?Max(-1,2),Max({^1986/12/10},{^2012/09/08})

运行结果：

2 09/08/12

(4) 最小值函数

函数格式：

Min(<表达式表>)

功能：返回表达式表中的最小值。若数据类型是日期时间型，则返回最前面的日期或时间。

【例 1-15】

?Min(-1,2),MAX({^1986/12/10},{^2012/09/08})

运行结果：

-2 12/10/86

(5) 绝对值函数

函数格式：

Abs(<数值表达式>)

功能：返回数值表达式的绝对值。

【例 1-16】

?Abs(-23), ABS(0)

运行结果：

23 0

2. 字符函数

(1) 求字符串长度函数



函数格式：

Len (<字符串表达式>)

功能：返回字符串表达式中字符的个数。空串的返回值是 0。

【例 1-17】

?Len(" 14 级 计算机应用专业 ")

运行结果：

19

(2) 大写转换成小写函数

函数格式：

Lower(<字符串表达式>)

功能：将字符串表达式中的大写字母全部转换成小写字母，其他字符不变。

【例 1-18】

?Lower("My name is 王红")

运行结果：

my name is 王红

(3) 小写转换成大写函数

函数格式：

Upper(<字符串表达式>)

功能：将字符串表达式中的小写字母全部转换成大写字母，其他字符不变。

(4) 删除尾部空格函数

函数格式：

RTrim(<字符串表达式>)

或者 Trim (<字符串表达式>)

功能：删除字符串表达式尾部空格后返回。

(5) 删除前导空格函数

函数格式：

LTrim(<字符串表达式>)

功能：删除字符串表达式首部空格后返回。

(6) 删除前导和尾部空格函数

函数格式：

AllTrim(<字符串表达式>)

功能：删除字符串表达式前部和后部的空格后返回。

【例 1-19】

?Len(" GOOD "),Len(AllTrim(" GOOD ")),Len(RTrim(" GOOD ")), Len(LTrim(" GOOD "))

运行结果：

7 4 6 5

(7) 取左边子串函数

函数格式：

Left(<字符串表达式>,<数值表达式>)

功能：返回从字符串表达式左边取出 N (N =数值表达式) 个字符和形成的子串。当 N 大于 Len(<字符串表达式>)时，返回整个字符串；若 N 为负数，则返回空串。

(8) 取右边子串函数

函数格式：

Right(<字符串表达式>,<数值表达式>)

功能：返回从字符串表达式右边取出 N (N =数值表达式) 个字符和形成的子串。

(9) 取子串函数

函数格式：

SubStr (<字符串表达式>,<数值表达式 1>,[<数值表达式 2>])

功能：从字符串表达式中第 N 位开始，取 M 个字符作为子串返回（其中 N =数值表达式 1， M =数值表达式 2）。当<数值表达式 2>省略时，子串从 N 位开始到最后一位。

【例 1-20】

Mystr="I feel good."

?Left(Mystr,4),Right(Mystr,5),SubStr(Mystr,3,4)

运行结果：

I fe good. feel

(10) 求子串的位置函数

函数格式：

At(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

Atc(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

功能：返回字符表达式 1 在字符表达式 2 里出现第 N 次的位置数 (N =数值表达式；当<数值表达式>省略时， $N=1$)。返回值为 0 时，表示未出现子串。At 函数执行时区分字符表达式的大小写，Atc 函数不区分。

【例 1-21】

?At("TH","THIS IS SOMETHING IMPORTANT."),At("TH","THIS IS SOMETHING
IMPORTANT.",2),At("TH","This is something important.",2)

运行结果：

1 13 0

(11) 求子串出现在最右边的位置函数

函数格式：

Rat(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

功能：返回一个字符表达式 1 在另一个字符表达式 2 内最后(最右边)出现的位置。 N =数值表达式，当 N 大于 1 时，返回从右边查找第 N 次时的位置。

(12) 字符串替换函数

函数格式：

Stuff(<字符串表达式 1>,<数值表达式 1>,<数值表达式 2>,<字符串表达式 2>)

功能：将从字符串表达式 1 中的位置 N 开始的 M 个字符用字符串表达式 2 替换（其中 N =数值表达式 1， M =数值表达式 2）。

【例 1-22】

?Stuff("计算机管理与维护",7,4,"安全")

运行结果：

计算机安全与维护

3. 日期时间函数

(1) 系统日期、时间和日期时间函数

函数格式：

Date()|Time()|DateTime()

功能：Date()返回当前系统日期，返回值为日期型常量，默认格式是“月/日/年”。Time()



返回当前系统时间, 返回值为字符型常量, 格式是“时:分:秒”。

DateTime()返回当前系统时间, 返回值为日期时间型常量。

【例 1-23】

```
?Date(), Time(), DateTime()
```

运行结果:

```
12/20/13    13:13:00    12/20/13  13:13:00 PM
```

(2) 求年、月、日函数

函数格式:

```
Year|Month|Day(<日期型表达式>/<日期时间型表达式>)
```

功能: 返回日期型或日期时间型参数中的年份、月份或日期值。

【例 1-24】

```
?Year(Date()), Month(Date()), Day(Date())
```

运行结果:

```
2013  12  20
```

(3) 求时、分、秒函数

函数格式:

```
Hour|Minute|Sec(<日期时间型表达式>)
```

功能: 返回日期时间型参数中的时、分、秒值。

【例 1-25】

```
?Hour(DateTime()), Minute(DateTime()), Sec(DateTime())
```

运行结果:

```
13   40   45
```

4. 数据类型转换函数

(1) 数值型转换成字符型函数

函数格式:

```
Str(<数值表达式 1>[, <数值表达式 2>[, 数值表达式 3]])
```

功能: 将数值表达式 1 中的值转换成 N 位字符串返回, 其中小数部分转换成 M 位 (N =数值表达式 2, M =数值表达式 3)。若 N 小于实际数值的整数位数, 则返回 N 个“*”号, 否则转换时多余位数补 0。若 M 省略, 则只转换整数; 若 N 省略, 则按 $N=10$ 处理, 字符串前补充相应位数的空格。

【例 1-26】

```
A=Str(123)
```

```
?A, Len(A), Str(34.68, 5, 1), Str(345.6, 2)
```

运行结果:

```
123    10    34.7    **
```

(2) 字符串型转换成数值型函数

函数格式:

```
Val(<字符串表达式>)
```

功能: 将字符串表达式中由数字、正负号及小数点组成的字符串转换成数值型数据并返回。转换时遇到非数字字符时停止, 若第一个字符为非数字字符, 则返回 0.00 (Visual FoxPro 系统默认小数点后有两位)。

【例 1-27】

```
?Val(".12"), Val("3E.1"), Val("D34")
```

运行结果：

```
0.12  3.10  0.00
```

(3) 字符串型转换成日期型函数

函数格式：

```
CTOD(<字符串表达式>)
```

功能：将字符串表达式转换成日期型数据返回。

【例 1-28】

```
?CTOD("12/13/13")
```

运行结果：

```
12/13/13
```

(4) 日期型转换成字符串型函数

函数格式：

```
DTOC(<日期表达式>[,1])
```

功能：将日期表达式转换成字符串型数据返回。有参数 1 时，返回格式为 YYYYMMDD，无参数 1 时，返回格式为 MM/DD/YY。

【例 1-29】

```
X=CTOD("12/13/13")
```

```
?DTOC(X), DTOC(X,1)
```

运行结果：

```
12/13/13  20131213
```

(5) 字符型转 ASCII 码函数和 ASCII 码转字符型函数

函数格式：

```
ASC(<字符串表达式>)|Chr(<数值表达式>)
```

功能：将字符串表达式中的首字符转换成对应的 ASCII 码并返回，或将数值表达式代表的 ASCII 码转换成对应的字符并返回。

【例 1-30】

```
?ASC("after"), Chr(65)
```

运行结果：

```
97  A
```

5. 测试函数

(1) 空值 Null 测试函数

函数格式：

```
IsNull(表达式)
```

功能：判断表达式的值是否为 NULL 值，如果是，则返回逻辑值.T，否则返回.F。

(2) “空”值测试函数 Empty

函数格式：

```
Empty(表达式)
```

功能：判断表达式的值是否为空，如果是，则返回逻辑值.T，否则返回.F。

代表各数据类型的空值规定如下。

字符型：空串、空格、制表符、回车符、换行符。

日期型或日期时间型：空，如 CTOD(“ ”)。

数值型、整型、浮点型、货币型：0。

逻辑型：假(.F.)。



通用型或备注型：空（没有内容）。

【例 1-31】

```
STORE .NULL. TO X
Mystr= " "
Num=0
?IsNull(X),Empty(Mystr),Empty(Num)
```

运行结果：

```
.T. .T. .T.
```

（3）数据类型测试函数

函数格式：

Type(<字符表达式>)

功能：测试字符表达式的数据类型，返回字符型数据。字符表达式需要加定界符（单引号或双引号），返回值所代表的含义如表 1-7 所示。

表 1-7 数据类型测试函数的返回值

返回字符	数据类型	返回字符	数据类型
C	字符型或备注型	G	通用型
N	数值型、整型、浮点型、双精度浮点型	D	日期型
Y	货币型	T	日期时间型
L	逻辑型	X	NULL
O	对象型	U	未定义

【例 1-32】

```
X= " 45 "
?Type('X')
```

运行结果：

```
C
```

（4）条件测试函数

函数格式：

IIF(<逻辑表达式>,<表达式 1>,<表达式 2>)

功能：测试逻辑表达式的值，若为真，返回表达式 1 的值；否则，返回表达式 2 的值。

【例 1-33】

```
?IIF(50>90,"TRUE","FALSE")
```

运行结果：

```
FALSE
```



项目小结

本项目以 Visual FoxPro 的基本数据及其运算方法作为知识点，运用在 Visual FoxPro 数据库管理系统中不同的工作任务引导大家逐步熟悉 Visual FoxPro 的运行环境并能够对基本数据运算进行熟练操作。



上机实习

1. 启动 Visual FoxPro, 在用户界面下显示并隐藏所有工具栏。
2. 通过命令清空工作区窗口信息, 并执行退出命令。
3. 在磁盘新建文件夹“STUDENT”, 并通过菜单项添加该路径为 Visual FoxPro 的搜索路径。
4. 使用输出命令“?”输出下列图形, 如图 1-47 所示。
5. 将上题中的命令输入到文件 MYDRAW.PRG 中, 并执行该程序。
6. 新建一个项目文件 STUDENT.PJX, 并将上题中的 MYDRAW.PRG 添加进来, 编译生成 STUDENT.EXE 文件。

```

      1
    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
  1 2 3 2 1
    1 2 1
      1
  
```

图 1-47 图形

7. 假设学生甲的出生日期是{^1994-12-30}, 计算他的年龄是多少天?
8. 输出下列字符串表达式的值。

```

"ap "+"ple "
"ap "-"ple "
"ap "$" red apple "
"ap"$" red apple "
"a p "$"a p p l e "
"a p"$"a p p l e "
  
```

9. 假设有两个变量, XB 存储的是学生性别, 数据类型为逻辑型, .T.表示男生, .F.表示女生; CJ 存储的是学生“计算机应用基础”课程成绩, 数据类型为数值型。若 XB=.T., CJ=98, 请列出表达式验证这个学生是“成绩大于 90 的女生”吗?
10. 调用常用函数: 将今天的日期转变成字符串, 最终输出该字符串的长度。
11. 将字符串“I am a Chinese student.”中的“Chinese”替换成“French”。
12. 字符“H”、“h”、“0”的 ASCII 码是多少? 求最大值和最小值。
13. 分别测试“12”、12、[^1998/12/12]、.Y.、{^2014-07-22}、\$12.2387、{^1998-09-21 12:34:10 am}的数据类型。



知识巩固

一、选择题

1. Visual FoxPro 的工作方式不包括 ()。
 - A. 执行应用程序
 - B. 执行命令
 - C. 使用菜单或工具栏
 - D. 数据备份和恢复
2. 退出 Visual FoxPro 用户界面的命令是 ()。
 - A. CLEAR
 - B. QUIT
 - C. CLOSE
 - D. RELEASE
3. 隐藏或显示工具栏的方法是 ()。
 - A. 在工具栏空白处右击, 选择隐藏或显示某项工具栏命令
 - B. 选择“显示”“工具栏”命令
 - C. CLOSE
 - D. RELEASE
4. Visual FoxPro 命令大致分为五大类, 命令“?”属于 ()。
 - A. 程序设计命令
 - B. 数据库操作命令
 - C. Visual FoxPro 环境
 - D. 输入输出命令



5. 下列对编辑命令的说法中, 错误的是 ()。
- A. “命令”对话框是编辑窗口, 可以在“命令”对话框中编辑、插入、删除、剪切、复制或粘贴文本
- B. 一条命令在书写时不能断开分行续写
- C. “命令”对话框里的文字可以改变字体、行间距等格式
- D. 命令行中的命令必须是半角状态下的字符
6. 按照变量命名规则, 下列变量中错误的是 ()。
- A. _A12 B. 13b C. B(12) D. 姓名
7. 设置系统显示日期型数据中的分隔符时使用 () 命令。
- A. SET CENTURY OFF B. SET MARK TO
- C. SET DATE YMD D. SET ECHO OFF
8. 对于数组的定义, 下列语句正确的是 ()。
- A. DIMENSION D(4,5) B. DIMENSION D(4,5) E(4,5)
- C. DIMENSION D(4 5) D. DIMENSION D(4,5)、E(4,5)
9. 假设 $C=[^1997/12/30]$, 那么命令 “?C” 的运行结果是 ()。
- A. ^1997/12/30 B. 12/30/97 C. 1997/12/30 D. 30/12/97
10. Visual FoxPro 的内存变量的数据类型不包括 ()。
- A. 数值型 B. 货币型 C. 日期时间型 D. 备注型
11. 表达式 “NAN”\$“BANANA” 的值是 ()。
- A. .T. B. .N. C. .F. D. “NANBANANA”
12. 假设 $X=“S”$, 执行命令 $?X=X+“UN”$ 的运行结果是 ()。
- A. .T. B. .N. C. .F. D. SUN
13. Empty(“ ”)和 IsNull(“ ”)的返回值分别是 ()。
- A. 都是.T. B. 都是.F. C. .T. .F. D. .F. .T.
14. 函数?Val(“4r”)的值是 ()。
- A. 4 B. 4.0 C. 4.00 D. 提示出错
15. 从字符串 “计算机应用基础” 中取出子串 “基础” 的表达式中, 错误的是 ()。
- A. SubStr(“计算机应用基础”, 11, 4) B. Right(“计算机应用基础”, 4)
- C. Right(“计算机应用基础”, 11, 4) D. SubStr(“计算机应用基础”, 11)
16. (1990-10-30) - (1990-10-28) + 6%4 的值是 ()。
- A. 0 B. 4 C. 1 D. 提示出错
17. {^1990-10-30} - {^1990-10-28} + 6%4 的值是 ()。
- A. 0 B. 4 C. 1 D. 提示出错
18. 在 Visual FoxPro 中, 关于项目管家的说法中, 错误的是 ()。
- A. 项目管家只能生成项目, 不能创建其他格式文档
- B. 项目管家可以将项目编译成应用程序 (APP), 也可以编译成可执行程序 (EXE)
- C. 通过项目管家可以调用各种设计器
- D. 通过项目管家可以调用各种创建向导
19. 删除指定内存变量的命令是 ()。
- A. CLEAR MEMORY B. RELEASE ALL
- C. RELEASE <内存变量表> D. RELEASE <内存变量表> [LIKE <通配符>]
20. 在项目管家中 “移去” 一个文件可以 ()。
- A. 在磁盘上删除该文件

- B. 删除该文件在该项目中的链接
 - C. 移动该文件在项目中的位置
 - D. 可以删除磁盘文件, 也可以从项目中移去该文件
21. 下列说法中, 错误的是()。
- A. 内存变量可以不定义直接赋值
 - B. 同一个内存变量可以多次赋给不同数据类型的值
 - C. 数组变量必须先进行定义再赋值
 - D. 数组变量也是内存变量, 可以直接赋值

二、填空题

- Visual FoxPro 的内存变量常用数据类型有_____6 种(用符号填写)。
- 给数组 SA(2,3)里的所有变量都赋值 100 的语句是_____。
- 要显示以“AB”开头的内存变量的命令是_____。
- 要编辑程序“main.prg”的命令是_____, 执行该程序的命令是_____。
- 命令 A=10, B=2, ?(5*A+-10)%(2^B)的输出结果是_____。
- 输入如下命令:

```
SET EXACT OFF
STORE "ABC" TO S1
STORE "ABC " TO S2
STORE "ABCDE" TO S3
?S1=S3, S3=S1, S1=S2, S2=S1, S2==S1
```

运行结果是_____, _____, _____, _____, _____。

```
SET EXACT ON
?S1=S3, S3=S1, S1=S2, S2=S1, S2==S1
```

运行结果是_____, _____, _____, _____, _____。

- 如果一个表达式中包含算术运算、关系运算、逻辑运算和字符运算, 则它们的运算优先级从高到低排列是_____, _____, _____, _____。
- 货币型数据类型的引导符是_____, 它的小数位数是_____位, 超过位数四舍五入。
- 字符型数据的边界符可以是_____, _____, _____。
- 字符串进行关系比较时是逐字符按照_____进行大小比较。

三、写出下列命令或表达式

- 给变量 STR1、STR2、STR3 赋值.T。
- 显示以 ST 开头的内存变量, 并将结果输出到文件 REPORT.txt 中。
- 显示今天的月份和日期。
- 释放当前内存中所有内存变量。
- 按照公式 $\left(\frac{5}{13} - \frac{2}{15}\right) \times 65.12$ 列出表达式, 并输出计算结果。
- 从字符串“D:\FOXPRO\A.PRG”中取出文件名字符串并输出。
- 定义一个有 10 个元素的一维数组 DIM, 并都赋初始值 0。
- 设置 MM#DD#YY 格式输出当前日期。
- 去掉字符串“You are successful!”末尾的空格。
- 写出编辑程序文件 CALL.PRG 的命令。



单元二

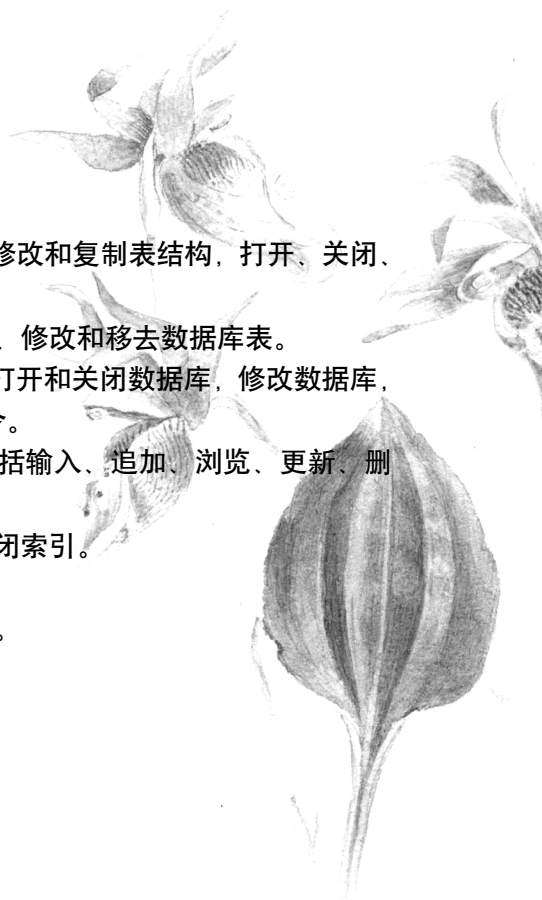
数据库及表的实现

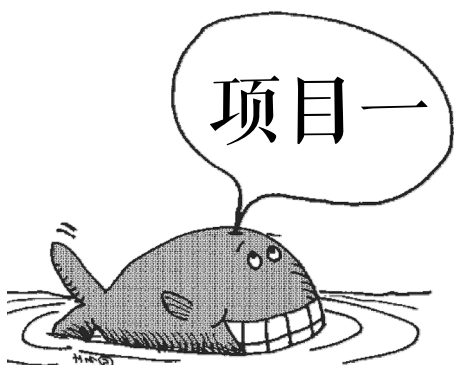
知识目标

- 1) 掌握自由表的概念、基本组成及命名规则。
- 2) 掌握表设计器、数据库设计器的功能。
- 3) 掌握 Visual FoxPro 命令的基本格式和使用方法。
- 4) 掌握数据库的基本概念及基本组成。
- 5) 理解数据库表和自由表的区别。
- 6) 掌握字段格式码和输入掩码的功能。
- 7) 掌握字段有效性规则和记录有效性规则的概念和功能。
- 8) 掌握索引的概念及分类。
- 9) 理解参照完整性的概念。

技能目标

- 1) 能使用不同方法创建自由表。
- 2) 熟练使用数据表操作命令，包括创建数据表，显示、修改和复制表结构，打开、关闭、删除数据表的命令。
- 3) 能够使用多种方法创建数据库、新建数据库表，添加、修改和移去数据库表。
- 4) 熟练使用数据库操作命令，包括创建和删除数据库，打开和关闭数据库，修改数据库，设置当前数据库，新建、添加数据库表，删除数据库表的命令。
- 5) 能使用菜单和命令两种方式管理数据表中的数据，包括输入、追加、浏览、更新、删除、恢复、排序、检索表数据。
- 6) 能使用命令方式建立索引、设置主控索引，打开和关闭索引。
- 7) 能建立数据表之间的临时关系并浏览多表间的数据。
- 8) 能建立数据表之间的永久关系并创建表间参照完整性。





创建学生信息表

项目说明

本项目介绍在 Visual FoxPro 数据库管理系统中实现关系数据模型的方法,内容包括数据表结构的创建和修改,实施过程包含 3 个任务。

任务一：使用表向导创建“学生自然情况表”。

任务二：使用表设计器创建“课程情况表”。

任务三：显示和修改“课程情况表”的结构。

任务一 使用表向导创建“学生自然情况表”

任务目标

能够使用表向导创建数据表结构。

任务描述

在 Visual FoxPro 环境下利用表向导创建学生自然情况表 Student.dbf。

相关知识

1. 字段的数据类型

一个 Visual FoxPro 数据表最多可以包含 255 个字段,可以通过指定字段名、字段数据类型、字段宽度、数值型或浮点型字段包含的小数位数、字段上的索引及字段是否接受 NULL 值这 6 个常规属性来描述字段。

字段名：字段名不允许重复。对于自由表,若字段名长度大于 10 个字符,Visual FoxPro 只存储前 10 个字符。字段名由汉字、字母或下画线开头,由汉字、字母、数字或下画线组成,



不允许有空格和其他字符出现，且不区分字母大小写。

字段数据类型：用来定义字段存储值的类型。根据不同字段所代表的意义设计不同的数据类型。常用的字段数据类型有字符型、数值型、日期型、货币型、逻辑型、备注型、通用型等。

字段宽度：定义字段存储值可以占用的最大字符数。表 2-1 描述了各种常用数据类型的字段宽度。可以设置字段宽度的数据类型是字符型、数值型、浮点型和双精度浮点型。

表 2-1 各种常用数据类型的字段宽度

字段数据类型	说 明	字 段 宽 度	小 数 位 数	实 例
字符型	可以是字母、汉字或数字等字符型文本，每个字符占 1 字节，每个汉字占 2 字节，最多可占 254 字节	$n \leq 254$		学生学号、姓名
货币型	货币值，小数点后保留 4 位	8		工资
日期型	由年、月、日组成的日期数据	8		出生日期
日期时间型	由年、月、日、时、分、秒组成的日期和时间数据	8		加班时间
整型	无小数数字的值	4		人数
数值型	整数或小数	$n \leq 20$	d	成绩
浮点型	需要精确计算的数值（整数或小数）	$n \leq 20$	d	面积
双精度型	精度更高、占用空间是浮点型存储值的两倍	$n \leq 20$	d	超小或超大的距离计算
逻辑型	取值为真（.T.）或假（.F.）	1		性别
通用型	可存储 OLE 对象的通用型字段类型	4		照片、简历
备注型	不定长的字母、数字、字符串	4		总结报告

小数位数：定义字段存储值中可以保留几位小数。数值型、双精度型、浮点型的小数位数不大于 20。字段宽度与小数位数的关系为

$$\text{字段宽度} = \text{整数位数} + 1 + \text{小数位数}$$

索引：定义字段是否为索引字段。除了备注型和通用型不能建立索引外，其他类型字段都可建立索引。

NULL：定义字段是否允许默认值或不确定值。数据表中尚未存储数据时，处于空值 NULL 状态。NULL 不等同于空格或零，不能与某个值进行比较。

2. 表文件命名规则

- 1) 数据表一般不能使用关键字作为文件名。
- 2) 不区分大小写。
- 3) 不能有下列符号：“？”、“\”、“/”、“*”、“|”、“>”、“<”、“ ”。
- 4) 一般取与表内容相关联的英文单词或汉语拼音缩写来命名。例如，学生成绩表可以使用 SCORE.DBF，也可以使用 XSCJ.DBF。
- 5) Visual FoxPro 数据表的扩展名为“.dbf”，当在数据表中设置了索引字段时，系统自动生成一个扩展名为“.cdx”的同名文件。当数据表中设置了通用型或备注型字段时，系统自动生成一个与所属数据表同名且扩展名为“.fpt”的同名文件。

任务分析

在单元一项目一中，设计了“学生成绩管理信息系统”的关系数据模型，其中对学生自然情况表的关系描述为：Student（学号，姓名，出生日期，性别，专业，籍贯，照片，父亲，母亲，联系电话，家庭住址）。

根据字段的属性确定 Student 表结构，如表 2-2 所示。

表 2-2 学生自然情况表 Student.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
学号	字符型	6			
姓名	字符型	8			
出生日期	日期型	8			
性别	字符型	2			
专业	字符型	30			
籍贯	字符型	10			
照片	通用型	4			
父亲	字符型	8			
母亲	字符型	8			
联系电话	字符型	11			
家庭住址	备注型	4			

下面将在 Visual FoxPro 中使用表向导创建该表结构。



实施步骤

1. 启动表向导

选择“文件”“新建”命令，在弹出的“新建”对话框中选中“表”单选按钮，单击“向导”按钮，如图 2-1 所示。

2. 字段选取

在弹出的“表向导”对话框中选取 11 个字段，并选择自由表的形式，表名为 Student.dbf，单击“下一步”按钮，如图 2-2 所示。



图 2-1 “新建”对话框

图 2-2 表向导设置第一步



3. 修改字段设置

根据设计好的学生自然情况表结构修改每个样本字段的名称、类型、宽度等属性，然后单击“下一步”按钮，如图 2-3 所示。

4. 索引表

以“学号”字段创建索引，勾选“学号”复选框，单击“下一步”按钮，如图 2-4 所示。



图 2-3 表向导设置第二步



图 2-4 表向导设置第三步

5. 完成

若表结构已经按需要设计完成，选中“保存表为以后使用”单选按钮，否则可以选中“保存表并在表设计器中修改它”单选按钮，单击“完成”按钮，如图 2-5 所示。



问题探究

1. 如何根据表文件名的扩展名判断是什么类型的文件？

Visual FoxPro 所使用的扩展名及其相关文件类型如表 2-3 所示。



图 2-5 表向导设置第四步

表 2-3 Visual FoxPro 所使用的扩展名及其相关文件类型

扩展名	文件类型	扩展名	文件类型	扩展名	文件类型
.dbf	表	.fpt	表备注	.mnx	菜单
.dbc	数据库	.tbk	备注备份	.mpr	已生产的菜单程序
.pjx	项目	.idx	单索引	.qpr	已生成的查询程序
.prg	程序	.cdx	复合索引	.scx	表单
.exe	可执行程序	.vue	视图	.frx	报表
.lbx	标签文件	.mem	变量存储	.app	已生成的应用程序

2. 如何判断某字段当前记录是否为 NULL 值？

NULL 应理解为没有任何值，不能等同于空格，也不等于零，判断当前记录的某个字段值是否为空值，表达式中需要使用常用函数 IsNull(字段名)，该函数的返回值为逻辑型，当字段值为 NULL 时，返回真(.T.)；否则，返回假(.F.)。

3. 创建本任务中的 Student.dbf 表后, 系统会同时保存几个文件?

因为学生自然情况数据表中包含了备注字段和索引字段, 所以保存时系统自动生成 3 个文件, 它们分别是 Student.dbf、Student.cdx、Student.fpt。

任务二 使用表设计器创建“课程情况表”

任务目标

能够使用表设计器创建数据表。

任务描述

在 Visual FoxPro 环境下利用表设计器创建课程情况表 Course.dbf。

任务分析

任务一讲解了如何利用表向导创建自由表, 其中最后一步操作中系统提示可以使用表设计器进行表结构修改。本任务即讲解如何使用表设计器创建课程情况表 Course.dbf。

课程情况表 Course.dbf 的关系描述为:

Course (课程编号, 课程名称, 学分, 授课学期, 授课老师, 学时, 考试)。

根据字段的属性确定 Course.dbf 的表结构, 如表 2-4 所示。

表 2-4 课程情况表 Course.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
课程编号	字符型	4			
课程名称	字符型	30			
学分	整型	4			
授课学期	字符型	1			
授课老师	字符型	8			
学时	整型	4			
考试	逻辑型	1			

下面将在 Visual FoxPro 中使用表设计器创建该表结构。



实施步骤

1. 新建数据表并指定表名及保存位置

选择“文件” “新建”命令。

在打开的“新建”对话框选中“表”单选按钮, 单击“新建文件”按钮。

在弹出的“创建”对话框中选择文件存放路径, 然后输入表名 Course.dbf, 单击“保存”按钮。如果没有输入文件扩展名, 系统将按照表的保存类型添加“.dbf”扩展名, 如图 2-6 所示。

2. 设计表结构

在表设计器中选择“字段”选项卡, 依次插入课程编号等字段, 设置每个字段的类型、宽



度、小数位数及索引和 NULL 属性，如图 2-7 所示。

设置“课程编号”为索引字段后，在“索引”选项卡中自动添加一条排序方式，按照“课程编号”字段升序排序，索引类型为普通索引，单击“确定”按钮后完成表结构设计，如图 2-8 所示。

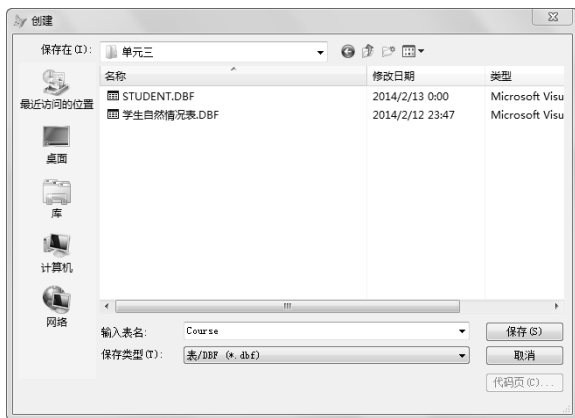


图 2-6 “创建”对话框

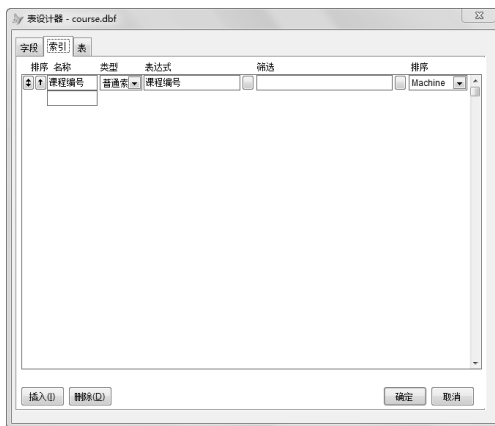


图 2-7 在表设计器中设置课程表结构



问题探究

1. 如何使用命令创建自由表？

除了使用表向导、表设计器创建自由表的方法外，还可以通过在“命令”对话框中执行命令快速创建数据表。

(1) 创建一个新数据表的命令

命令格式：

```
CREATE [<表文件名>|?]
```

功能：通过表设计器创建表。

参数：

[<表文件名>|?]: 可选项。表文件名可以包含路径，当扩展名“.dbf”省略时，系统默认其扩展名为“.dbf”。若表文件名省略或输入“?”时，将弹出“创建”对话框，指定要创建的表名及保存位置后才能打开表设计器创建表，如图 2-9 所示。



图 2-8 创建索引字段

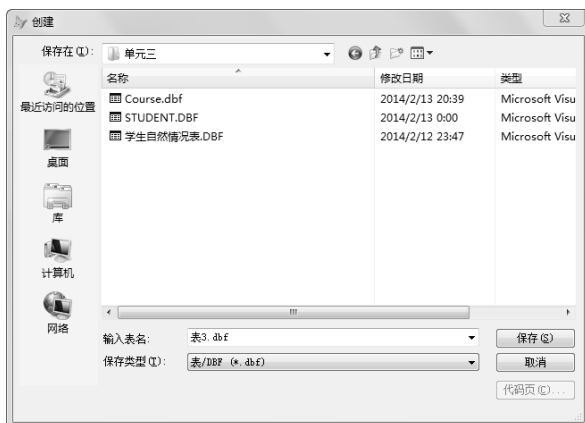


图 2-9 “创建”对话框

(2) 通过复制表结构创建数据表的命令

命令格式：

```
COPY STRUCTURE TO <新表文件名> [FIELDS 字段 1, 字段 2.....]
```

功能：创建一个与当前已打开的表结构相同的新表，且是空白表。

参数：

<新表文件名>：必选项。新表文件名不能与已有表同名。

[FIELDS 字段 1, 字段 2.....]：可选项。FIELDS 后接需要复制的字段列表，使用逗号相隔。该参数若省略，表示复制全部字段。

2. 创建数据表的方式有哪几种？

使用表向导创建。

使用表设计器创建。

使用命令创建。

在项目管理器中创建。

任务三 显示和修改“课程情况表”的结构

任务目标

能够显示并修改数据表的结构。

任务描述

创建了课程情况表 Course.dbf 后，有时需要根据实际情况对表结构进行修改，通过本任务的实施，浏览原表结构并增加、删除、修改原字段，修改后的表结构如表 2-5 所示。

表 2-5 修改后课程情况表 Course.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
课程编号	字符型	4			
课程名称	字符型	30			
学分	整型	4			
授课学期	字符型	1			
参考学时	整型	4			
考试	逻辑型	1			
推荐教材	通用型	4			

其中，需要修改“参考学时”字段，添加“推荐教材”字段，删除原有字段“授课老师”。

相关知识

1. 表文件的打开

表文件在创建完成后保存在外存中，如果需要修改，要先将它调入内存，即打开数据表后，方能显示或修改。



Visual FoxPro 提供了多种方式打开数据表，常见的方式如下。

（1）使用菜单打开数据表

选择“文件”“打开”命令，如图 2-10 所示，在弹出的“打开”对话框中选择需要打开的数据表，单击“确定”按钮，如图 2-11 所示。



图 2-10 选择“打开”命令

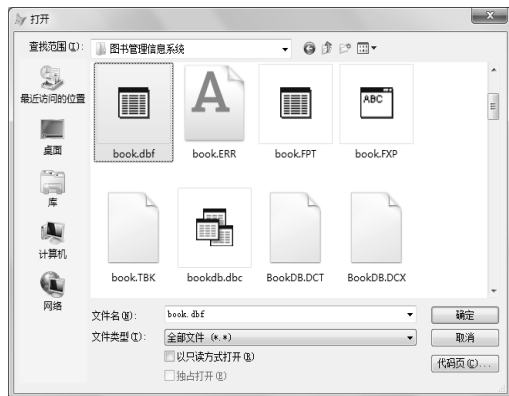


图 2-11 “打开”对话框

（2）使用命令打开数据表

命令格式：

USE [表文件名|?]

功能：打开或关闭数据表。

参数：

[表文件名|?]: 可选项。带表文件名参数时，为打开该数据表，省略时为关闭当前数据表。若不指定表文件名而使用“？”，系统会弹出“使用”对话框，如图 2-12 所示，以便指定需要打开的数据表。打开数据表的同时，同名的“.fpt”和“.cdx”文件也将自动打开。

（3）使用项目管理器打开数据表

在项目管理器中选中需要打开的表，单击“修改”或“浏览”按钮即可直接打开数据表，如图 2-13 所示。

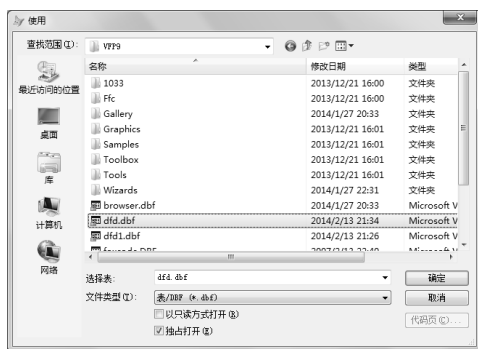


图 2-12 “使用”对话框



图 2-13 使用项目管理器打开或修改数据表

关闭“浏览”或“修改”对话框后，该表仍然处于打开状态。

2. 表文件的关闭

数据表在使用完后,通过关闭表文件将该表从内存空间中释放。一般使用不加表名的 USE 命令来关闭当前打开的表文件。

3. 表文件的删除

命令格式:

DROP TABLE <表文件名>

功能:从磁盘上删除表文件。

参数:

<表文件名>:必选项。表文件名可以包含路径。若该参数省略,执行时系统将弹出“删除”对话框,提示选择需要删除的表文件后,单击“删除”按钮,如图 2-14 所示。

任务分析

表结构建立后,有时需要更改字段名称、增加或删除字段、修改字段属性、更改索引,这些操作仍然需要在表设计器中完成,但在操作前必须先打开表。



实施步骤

1. 显示表结构

在“命令”对话框中输入以下命令,显示 course.dbf 表结构。

USE Course &&打开 Course.dbf 数据表,表文件名可以包含路径
LIST STRUCTURE 或者 DISPLAY STRUCTURE &&显示表结构的命令

显示结果包含文件路径、文件名、记录数、最新更新时间以及各字段属性,如图 2-15 所示。

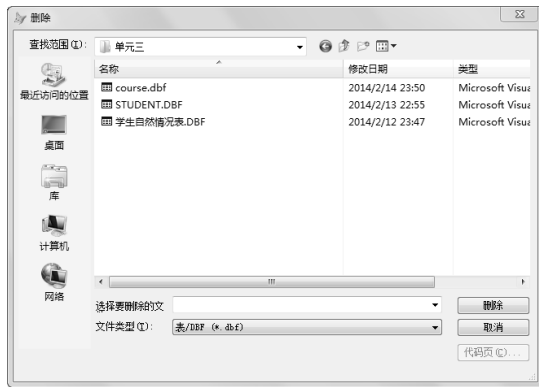


图 2-14 “删除”对话框

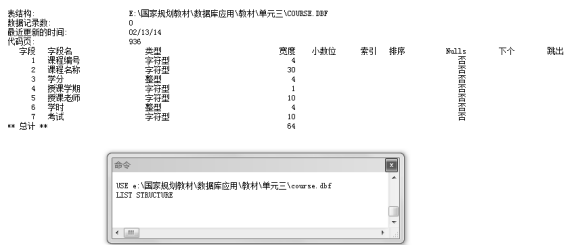


图 2-15 课程表 course.dbf 表结构显示结果

2. 修改表结构

打开表设计器修改表结构有两种方法。

在项目管理器中修改表结构。在数据选项卡中选择 course 数据表,再单击“修改”按钮,打开表设计器进行修改,如图 2-16 所示。

使用命令进行修改,如图 2-17 所示。

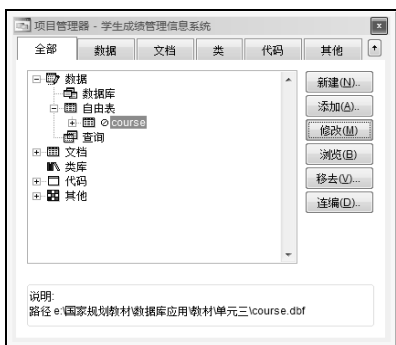


图 2-16 在项目管理器中修改表结构

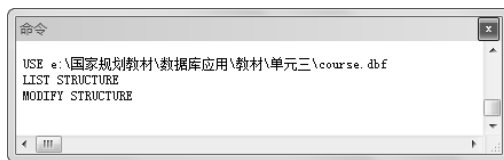


图 2-17 使用命令修改表结构

命令格式：

MODIFY STRUCTURE

功能：修改当前已打开表的结构。

使用任何一种方法在表设计器中打开数据表 course 后，对表结构做如下修改。

更改“学时”字段的名称为“参考学时”。

选择“授课老师”字段，单击“删除”按钮。

在“考试”字段后再添加“推荐教材”字段，类型为“通用型”，勾选“NULL”复选框。最后单击“确定”按钮，保存修改并退出表设计器，如图 2-18 所示。



问题探究

1. 如何更改字段类型和宽度？

在更改字段类型和宽度时要防止数据丢失。如果数据表内还没有存储记录，字段的类型和

宽度可以随意更改，但如果数据表内按照原类型已输入记录，在修改字段类型后，可能系统并不完全转换字段的内容，或者根本不转换。例如，将数值型字段转换成货币型时，原数据将转换成小数位为 4 位的货币型，将日期型字段转换成逻辑型、货币型、备注型时，原数据将丢失。

此外，同类型数据在宽度变化时，原数据的精度将会发生变化，数据也会改变。

2. 如果修改表文件时发生错误，且更改结果已被保存，如何恢复成原表？

在修改表文件时，系统自动为表文件 (.dbf) 和它的备注文件 (.fpt) 分别生成一个同名且扩展名为 “.bak” 和 “.tbk” 的备份文件。若修改后发现错误，可恢复这两个文件的扩展名成原表。



图 2-18 修改后的课程表结构



项目小结

本项目通过创建学生自然情况表和课程情况表，介绍了创建和修改自由表的方法和命令。各自由表独立存储，之间没有建立任何联系。学习本项目的目的是为创建数据库做好准备。



上机实习

1. 创建学生选修课程表 Select.dbf, 表结构如表 2-6 所示。

表 2-6 学生选修课程表 Select.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
学号	字符型	6			
课程编号	字符型	4			
授课学期	字符型	1			
所得学分	整型	4			
期中成绩	数值型	5	1		
期末成绩	数值型	5	1		
总成绩	数值型	5	1		

2. 创建图书表 Book.dbf, 表结构如表 2-7 所示。

表 2-7 图书表 Book.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
图书编号	整型	4			
书名	字符型	50			
数量	整型	4			
价格	数值型	5	1		
存放位置	字符型	50			
ISBN 号	字符型	20			
出版社	字符型	20			
作者	字符型	50			

3. 创建读者表 Loander.dbf, 表结构如表 2-8 所示, 并复制表结构到新表 Reader.dbf。

表 2-8 读者表 Loander.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
读者编号	整型	4			
姓名	字符型	50			
出生年月	日期型	8			
单位	字符型	30			
职务	字符型	30			
电话	字符型	11			
教育程度	字符型	10			
证件号码	字符型	18			
照片	通用型	4			



4. 创建读者借书表 Loand.dbf, 表结构如表 2-9 所示。

表 2-9 读者借书表 Loand.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL
借书编号	整型	4			
读者编号	整型	4			
图书编号	整型	4			
借书时间	日期型	8			
还书时间	日期型	8			
续借	逻辑型	1			
滞纳金	货币型	8			

5. 在图书表 Book.dbf 表结构的基础上做如下修改。

- (1) 将字段“作者”调整到“书名”字段的后面, 字段“存放位置”调整到最后。
- (2) 在“存放位置”字段前新增“封面”字段, 设置为通用型。
- (3) 修改“出版社”字段类型的宽度为 40。
- (4) 修改“价格”字段的名称为“购买价格”, 数据类型为“货币型”。



知识巩固

一、选择题

1. 一个数据表最多可以包含 () 个字段。
A. 256 B. 128 C. 20 D. 255
2. 根据字段名的命名规则, 下面 () 字段名是正确的。
A. 姓名 B. 3NAME C. -3NAME D. /3 姓名
3. 不可以求和计算的数据类型是 ()。
A. 字符型 B. 数值型 C. 整型 D. 浮点型
4. 学生成绩表包含“总成绩”字段, 类型为浮点型, 字段宽度为 6, 小数位数 1, 那么整数位数最大是 () 位。
A. 5 B. 4 C. 6 D. 3
5. 下列字段类型中不能作为索引字段的类型是 ()。
A. 日期型 B. 字符型 C. 逻辑型 D. 备注型
6. 自由表的文件扩展名是 ()。
A. .dbf B. .fpt C. .cdx D. .tbk
7. 不论使用哪种方法创建表, 都会打开 () 对话框进行设计。
A. 表向导 B. 命令窗口 C. 表设计器 D. 文件选择
8. 通常嵌入或存储 OLE 对象的字段类型是 ()。
A. 备注型 B. 字符型 C. 逻辑型 D. 通用型
9. 下列说法中, 正确的是 ()。
A. 一个包含备注字段的自由表在保存时只生成一个表文件, 该文件的扩展名为.dbf
B. NULL 值等同于空格
C. 在修改表结构时, 字段的类型和宽度都能够修改

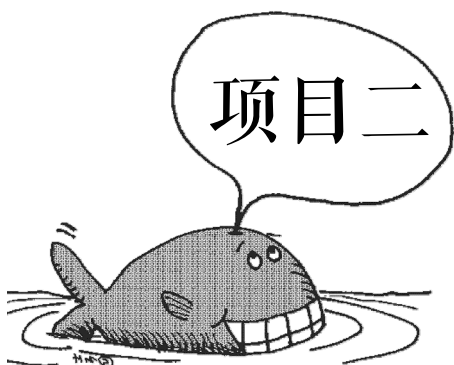
- D. 无论使用什么方法修改表结构, 该表都自动被打开
10. 修改表结构的命令是()。
- A. DISPLAY STRUCTURE B. LIST STRUCTURE
C. MODIFY D. MODIFY STRUCTURE
11. 在 Visual FoxPro 中, 逻辑型、通用型、日期型字段的宽度由系统自动给出, 它们分别是() 字节。
- A. 1、4、8 B. 4、4、10 C. 1、10、8 D. 2、8、8
12. Visual FoxPro 的字段支持如下数据类型()。
- A. 字符型、数值型、二进制数值型、二进制字符型等
B. 字符型、数值型、通用型、图像型等
C. 字符型、数值型、日期型、时间型等
D. 字符型、数值型、日期时间型、备注型等

二、填空题

1. 一个自由表最多可以包含_____个字段, 它的字段名最多可以包含_____位字符。
2. 创建自由表的命令是_____。
3. 在创建数据表时如果定义了索引字段, 则在保存表文件时除了保存一个扩展名为_____的表文件, 同时还会自动生成一个扩展名为_____的索引文件。
4. 表结构可以通过命令复制, 该命令是_____。
5. 字符型字段的最大宽度是_____, 数值型字段的最大宽度是_____, 日期型字段的宽度是_____, 逻辑型字段的宽度是_____, 通用型字段的宽度是_____。

三、简答题

1. 以创建图书表 Book.dbf 为例, 简述 4 种创建自由表的方法及步骤。
2. 使用命令完成下列操作。
- (1) 打开图书表 Book.dbf。
- (2) 浏览表结构。
- (3) 将表结构复制到新表 Book_new.dbf。
- (4) 修改新表 Book_new.dbf 结构。
- (5) 关闭所有表。



创建“学生成绩管理”数据库

项目说明

数据库是存储在计算机存储设备上相关数据的集合，它包含相关数据表及表之间的关系。Visual FoxPro 数据库管理系统为创建、维护数据库提供了可视化工具——数据库设计器，数据库可以包含数据表、视图、存储过程和创建连接。用户可以单独创建和使用数据库，也可以在项目中添加数据库。

本项目包括 4 个任务。

任务一：创建“学生成绩管理”数据库。

任务二：添加自由表。

任务三：新建数据库表。

任务四：修改数据库。

任务一 创建数据库

任务目标

能够使用多种方法创建数据库。

能够熟练打开、关闭、删除、修改数据库及设置当前数据库。

任务描述

创建“学生成绩管理”数据库文件 StudentDB.dbc，保存在指定文件夹下。再创建“学生学籍管理”数据库文件 StdManageDB.dbc。设定当前数据库为 StudentDB.dbc。修改当前数据库 StudentDB.dbc。关闭所有数据库。删除“学生学籍管理”数据库文件 StdManageDB.dbc。



实施步骤

1. 创建数据库文件 StudentDB.dbc

创建“学生成绩管理”数据库文件 StudentDB.dbc，保存在指定文件夹下。

有 3 种方法创建数据库文件：

(1) 使用项目管理器创建数据库。

在项目管理器中打开 Student.pjx 项目文件。选择“数据”选项卡中的“数据库”选项。单击“新建”按钮，弹出“新建数据库”对话框，如图 2-19 所示。



图 2-19 使用项目管理器创建数据库

单击“新建数据库”按钮，弹出“创建”对话框，如图 2-20 所示。按照要求选择文件保存路径，输入数据库名，选择保存类型，然后单击“保存”按钮。

系统自动打开数据库设计器，此时在菜单中会新增“数据库”菜单项。打开数据库设计器工具栏，即可往数据库中添加内容（新建立的数据库是空的），如图 2-21 所示。

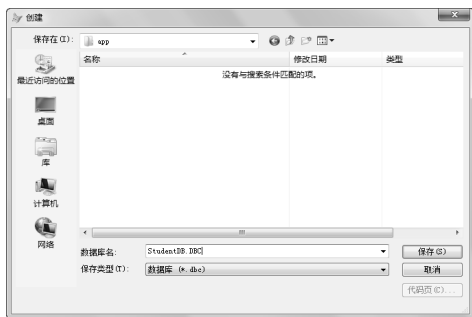


图 2-20 “创建”对话框

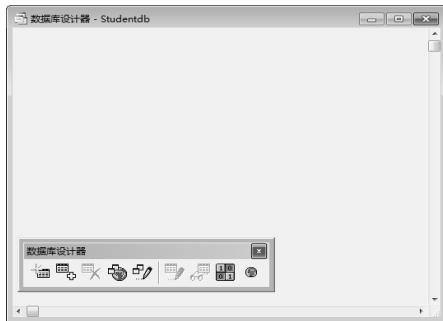


图 2-21 数据库设计器

(2) 使用菜单或工具栏创建数据库。

选择“文件” “新建”命令或单击“常用”工具栏中的“新建”按钮，弹出“新建”对话框。

选中“数据库”单选按钮，单击“新建文件”按钮，弹出“创建”对话框。

选择正确的保存路径，输入数据库文件名 StudentDB.dbc 后，单击“保存”按钮后，系统自动在数据库设计器中打开数据库。

(3) 使用命令创建数据库。

命令格式：

```
CREATE DATABASE [<带路径的库文件名>]
```



功能：创建数据库。

在“命令”对话框中执行如下命令：

```
CREATE DATABASE E:\学生成绩管理信息系统\DATA\StudentDB
```

该命令创建指定路径下的数据库（根据库文件的实际存储位置书写）。命令执行后，系统自动在库文件名后添加扩展名“.dbc”并保存在指定路径下，同时系统还会产生其他两个同名附加文件，其中“.dbt”是数据库备注文件，“.dcx”是数据库索引文件。使用命令方式创建数据库时，系统不出现数据库设计器。

2. 创建数据库文件 StdManageDB.dbc

创建“学生学籍管理”数据库文件 StdManageDB.dbc。执行如下命令：

```
CREATE DATABASE E:\学生成绩管理信息系统\DATA\StdManageDB
```

3. 设置数据库

设置当前数据库为 StudentDB.dbc。

在同一时刻可以打开多个数据库，但只能有一个数据库被操作，该数据库即为当前数据库。命令格式如下：

```
SET DATABASE TO [数据库文件名]
```

功能：设置当前数据库。

说明：该命令中的数据库必须是已经打开的数据库。

此时已经打开的数据库有两个：StudentDB.dbc 和 StdManageDB.dbc。输入如下命令：

```
SET DATABASE TO StudentDB
```

或者在“常用”工具栏中的“当前数据库”下拉列表中选择，如图 2-22 所示。

4. 关闭所有数据库

关闭数据库的方法有两种。

(1) 在项目管理器中关闭数据库

在项目管理器中依次选择数据库 StudentDB、StdManageDB，单击“关闭”按钮。

(2) 使用命令关闭数据库

命令格式：

```
CLOSE DATABASE [ALL]
```

功能：关闭当前打开的数据库和包含的数据库表。

5. 删除数据库文件

删除“学生学籍管理”数据库文件 StdManageDB.dbc。

删除数据库的方法也有两种。

(1) 在项目管理器中删除数据库

在项目管理器中选择数据库 StdManageDB，单击“移动”按钮，弹出如图 2-23 所示的对话框。



图 2-22 在“常用”工具栏中设置“当前数据库”



图 2-23 “删除”数据表对话框

单击“移去”按钮，将从项目中移去该数据库的链接；单击“删除”按钮，将从磁盘中删除该数据库。

(2) 使用命令删除数据库

命令格式：

```
DELETE DATABASE <数据库名|?> [DELETE TABLES] [RECYCLE]
```

功能：删除数据库文件及其表文件。

参数：

<数据库名|?>：必选项。若没有提供数据库名，可以输入问号“？”，系统将弹出“删除”对话框，如图 2-24 所示。

[DELETE TABLES]：可选项。同时删除数据库中包含的表文件。

[RECYCLE]：可选项。将被删除的数据库及数据表放入回收站。

在“命令”对话框中执行如下命令：

```
DELETE DATABASE E:\学生成绩管理信息系统\DATA\StdManage.dbc
```

说明：

- 若参数中的数据库文件名不带路径，那么系统就会在默认路径中查找该文件。
- 删除数据库前，该数据库必须是关闭状态。



问题探究

1. 使用命令创建数据库时若已存在同名数据库，系统将会如何处理？

Visual FoxPro 系统提供环境参数 SAFETY 用来标识当改写已有文件时 Visual FoxPro 是否出现提示的状态。该参数可以使用系统设置命令 SET SAFETY ON/OFF 来改变状态值，当环境参数 SAFETY 为“ON”时，指定在改写已有文件之前，显示“文件是否改写”的提示对话框；当参数为“OFF”时，则不提示任何信息，直接覆盖同名数据库。

2. 数据库关闭后，再次修改前如何打开该数据库？

通过打开项目管理器，选择“数据”选项卡中的相关数据库后单击“修改”按钮，可以在数据库设计器中直接打开该数据库。

通过选择“文件”“打开”命令，打开数据库设计器，同时打开该数据库。

通过命令打开数据库。

命令格式：

```
OPEN DATABASE [文件名]
```

功能：打开指定数据库。

说明：该命令中的文件名应是带路径的数据库文件名。若不带路径，则系统在默认路径中查找该数据库文件。

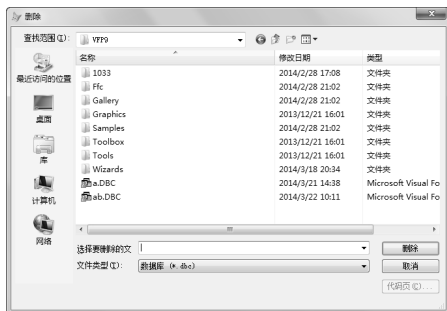


图 2-24 “删除”对话框

任务二 添加自由表

任务目标

能够熟练将自由表添入数据库。



能够熟练移去数据库表，将其转换成自由表。

任务描述

在数据库 StudentDB.dbc 中依次添加自由表：学生自然情况表 Student.dbf、课程情况表 Course.dbf 及学生选修课程表 Select.dbf。在该数据库中移去选修课程表 Select.dbf。

相关知识

独立的数据表即为自由表，当数据表被包含在数据库中时，被称为数据库表。一张数据表只能属于一个数据库，并且可以通过“移去”操作转换成为自由表。

数据库表与自由表在字段名长度、字段格式及有效性规则等方面有所区别。字段格式及有效性规则将在任务三中介绍。



实施步骤

1. 添加自由表

在数据库 StudentDB.dbc 中依次添加 3 个自由表：学生自然情况表 Student.dbf、课程情况表 Course.dbf 及学生选修课程表 Select.dbf。

在数据库中添加自由表有 3 种方法。

(1) 在项目管理器中添加表

使用项目管理器打开“学生成绩管理信息系统”项目文件 Student.pjx。

在“数据”选项卡中选择数据库 StudentDB，并单击其左侧图标旁的符号“+”，展开该数据库内容选项，展开状态下符号“+”变成了“-”，如图 2-25 所示。

选择“表”选项，单击“添加”按钮，在弹出的对话框中选择表名，如图 2-26 所示。



图 2-25 在项目管理器中打开数据库表



图 2-26 选择表名

依次在存储位置中选择自由表 Student.dbf、Course.dbf、Select.dbf，单击“确定”按钮。添加自由表后的项目管理器如图 2-27 所示。当自由表被添加到数据库后，就成为了数据库表。

(2) 在数据库设计器中添加表

打开数据库设计器，选择数据库“添加表”命令，依次添加自由表 Student.dbf、Course.dbf、Select.dbf，如图 2-28 所示。或者单击“数据库设计器”工具栏中的“添加表”按钮，如图 2-29

所示。

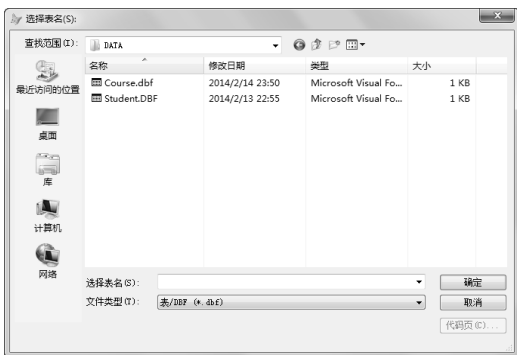


图 2-27 在项目管理器添加数据库表

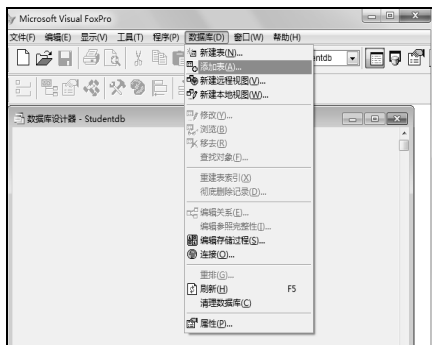


图 2-28 在数据库设计器中添加数据库表

(3) 使用命令添加表

命令格式：

ADD TABLE [表文件名]

功能：在当前打开的数据库中添加一个自由表。

在“命令”对话框中依次执行如下命令：

```
OPEN DATABASE StudentDB
ADD TABLE Student
ADD TABLE Course
ADD TABLE Select
```

2. 从数据库中移去选修课程表 Select.dbf

从数据库中移去数据表的方法也有三种。

(1) 在项目管理器中移去数据库表

在项目管理器“数据”选项卡中展开数据库 StudentDB 的“表”选项，如图 2-30 所示，选择数据表 Select，单击“移去”按钮，弹出如图 2-30 所示的对话框。



图 2-29 “数据库设计器”工具栏中的“添加表”按钮



图 2-30 在项目管理器中移去数据库表

单击“移去”按钮，将从数据库中移去表 Select 的链接；单击“删除”按钮，将从磁盘中删除该数据表。

(2) 在数据库设计器中移去表

在数据库设计器中打开数据库 StudentDB，选择需要移去的数据表 Select，使该表的标题栏呈高亮显示，然后选择“数据库”“移去”命令，如图 2-31 所示，弹出如图 2-32 所示的“移去”对话框。或者通过“数据库设计器”工具栏中的“移去表”按钮移去数据表 Select，如图 2-33 所示。

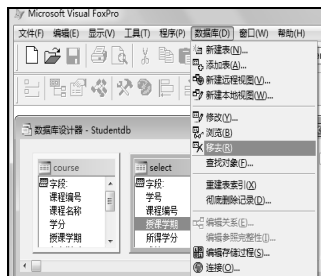


图 2-31 在数据库设计器中移去数据表



图 2-32 “移去”对话框

(3) 使用命令移去表

命令格式：

```
REMOVE TABLE [表文件名|?] [DELETE]
[RECYCLE]
```

功能：从当前打开的数据库中移去一个数据库表。

参数：

[表文件名|?]：可选项。若命令中没有提供表文件名，或输入参数为问号“？”时，系统将弹出“移去”对话框。

[DELETE]：可选项。从数据库中移去数据表，并且从磁盘上删除该表。

[RECYCLE]：可选项。将被删除的数据表文件放入回收站。

在“命令”对话框中执行如下命令：

```
OPEN DATABASE StudentDB
REMOVE TABLE Select
```



图 2-33 “数据库设计器”工具栏中的“移去表”按钮



问题探究

1. 在 Visual FoxPro 数据库中 can 添加哪些对象？如何理解“移去数据表”的含义？

当创建数据库时，Visual FoxPro 以独占方式创建并打开一个数据库（.dbc）文件。这个数据库像一个文件集合，可以包含数据表、视图、存储过程、数据连接等对象，也可以提供建立表间的永久关系、创建字段和记录有效性规则、创建维护表间数据一致性的规则（参照完整性）等操作。在数据库文件中存储数据表，实际上存储的是数据表文件的路径指针。“移去数据表”的操作实际上是删除该表在数据库中的路径指针，但并没有删除存储在磁盘上的数据表文件。

2. 自由表和数据表在字段名长度上有何区别？

自由表在创建时，字段名最多包含 10 个字符，而数据库表的字段名最多可以包含 128 个字符。当自由表转换为数据库表时，字段名不会发生改变，但是当数据库表从库中移出变成自由表时，长字段名就会被截为由前 10 个字符组成的字段名存储在表文件中。

任务三 新建数据库表

任务目标

掌握格式码和输入掩码的功能。

掌握字段有效性规则和记录有效性规则的概念和功能。

能够根据字段属性熟练设置数据库表字段的格式、显示标题和输入掩码。

能够根据字段属性，利用表达式生成器生成字段有效性规则和记录有效性规则。

任务描述

在数据库 StudentDB.dbc 中新建学生选修数据库表 SelCourse.dbf，并按要求创建不同字段的显示格式和有效性规则，具体要求如表 2-10 所示。

表 2-10 数据表 SelCourse.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	显示	字段有效性	记录有效性
学号	字符型	6		格式：T 输入掩码： 999999 标题： 学生学号	默认值： SUBSTR(STR(YEAR(DATE())),4,3,2) + " 9999 "	规则： 所得学分<= 课时/18 信息： " 所得学分 的满分为该门 课程的总课时 /18。 "
课程编号	字符型	4		输入掩码： X999		
授课学期	字符型	1		格式：T 输入掩码：9	规则：授课学期 => " 1 " .AND.授 课学期<=" 6 " 信息：" 授课学 期分别为 ' 1 '、 ' 2 ' ... ' 6 '，分别 代表 6 个学期 默认值：" 1 "	
所得学分	整型	4			规则：所得学分>=0	
课时	整型	4				
期中成绩	数值型	5	1		规则：成绩>=0 and 成绩<=100 信息：" 成绩应 该为小于等于 100 的正数。 "	
期末成绩	数值型	5	1		规则：成绩>=0 and 成绩<=100 信息：" 成绩应 该为小于等于 100 的正数。 "	
总成绩	数值型	5	1	标题： 平均成绩	规则：成绩>=0 and 成绩<=100 信息：" 成绩应 该为小于等于 100 的正数。 "	



相关知识

在新建数据库表时，Visual FoxPro 会新增一些自由表不具备的功能，主要包括以下功能。

设置字段内容的显示格式。

设置字段有效性规则。

设置记录有效性规则。

下面介绍在创建数据库表时的几个常用设置，如图 2-34 所示。

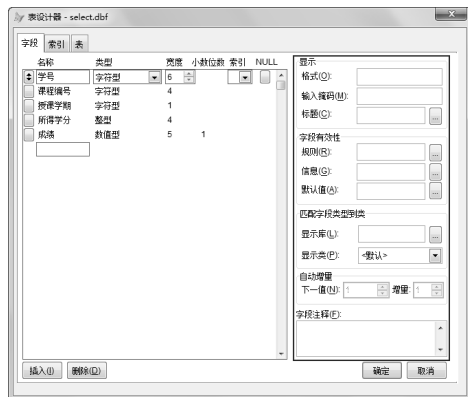


图 2-34 数据库表结构格式及有效性设置

1. 设置字段内容的显示格式

格式。“格式”功能是控制字段值在浏览窗口、表单、报表中的显示样式。格式码不区分大小写，应用于字段值中的所有字符，并且可以组合使用。常用的格式码及其功能如表 2-11 所示。

表 2-11 常用的格式码及功能

格式码	功能	适用数据类型
A	字段由字母字符组成，不允许有空格和标点符号	字符型
R	显示文本框的格式掩码，但不保存到字段中	所有类型
D	使用当前 SET DATE 格式	日期型或日期时间型
T	删除前导空格和结尾空格	字符型
E	英国日期格式	日期型或日期时间型
!	字母字符转换成大写	字符型
K	光标移至该字段并选择所有内容	所有类型
^	用科学计数法表示数值数据	数值型
L	数值字段显示前导 0	数值型
\$	显示货币符号	数值型

字段的显示格式如图 2-35 所示。

输入掩码：控制输入该字段的数据格式。一个掩码字符代表一个数据位。常用掩码字符及功能如表 2-12 所示。



图 2-35 字段的显示格式

表 2-12 常用掩码字符及功能

掩码字符	功 能	适用数据类型
X	任意字符位	字符型
*	左侧显示“*”	所有类型
9	代表一个数字字符位、“+”、“-”号位	数值型或字符型
.	代表一个小数点位	数值型和字符型
#	代表一个数字字符位、“+”、“-”号位和空格位	数值型和字符型
,	使用逗号分隔整数部分的逗号位	数值型和字符型
\$	指定显示货币符号位	数值型和字符型
\$\$	指定显示货币符号位，货币符号和数字符号不分开显示	数值型和字符型

标题：在数据库表中，可以在浏览窗口或表格栏标头中指定并显示一个标题来代替字段名。例如，若表结构中字段名用的是英文，要想使显示的字段标题为汉字，可以在标题栏中输入要显示的汉字字段名，这样显示该字段值时就比较直观了。如果没有设置标题，则将表结构中的字段名作为该字段的标题。

2. 设置字段有效性规则和记录有效性规则

字段有效性和记录有效性规则：限定可以输入到字段数据的有效范围。确定了字段有效性或记录有效性规则后，当字段值改变时，Visual FoxPro 系统就会自动依据设置好的该字段有效性规则和记录有效性规则检查输入的内容是否有效，如图 2-36 所示。

有效性规则的创建即生成有效性表达式，在表达式生成器中构建。生成包含字段变量的表达式，用来限定字段值的有效范围，如图 2-37 所示。例如，对“年龄”字段的有效性规则可以设置为：



图 2-36 字段有效性及记录有效性设置



图 2-37 表达式生成器

年龄 =>0 .AND. 年龄<=100

记录有效性规则是可以使用多个字段变量的表达式，用来确定数据输入范围。例如，通过记录有效性规则可以确定一个字段中的值始终大于同一记录中的另一个字段的值。

在当前字段发生改变时，启动字段有效性检查；在当前记录发生改变时，启动记录有效性检查。



信息：当向设置了有效性规则的字段输入不符合规则的数据时，就会显示“信息”文本框中的文字内容。

默认值：默认值也是一个表达式。它是系统在该字段中预置的值。例如，在“性别”字段的“默认值”文本框中输入字符常量“男”，那么在输入记录时只有女生才需要改变默认值，这样可以减少输入信息量。

以上功能是数据库表在创建时可以设置的属性，自由表在创建时，这些功能不可选。



实施步骤

1. 打开数据库

打开在任务一中创建的数据库 StudentDB.dbc。

选择“文件”“打开”命令，弹出“打开”对话框，如图 2-38 所示。

查找存放路径，选择在任务一中创建的数据库文件 StudentDB.dbc，单击“确定”按钮，打开数据库设计器，如图 2-39 所示。也可以通过打开项目管理和命令的方式打开数据库，在这里不多描述。

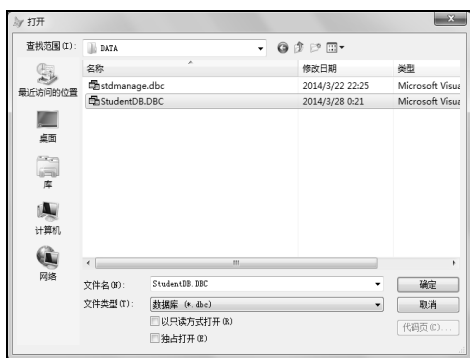


图 2-38 “打开”对话框



图 2-39 打开数据库设计器

2. 新建数据库表，指定表名及保存位置

当数据库打开时，新建的数据表都是数据库表，且都包含在该数据库中。

单击“数据库设计器”工具栏中的“新建表”按钮，如图 2-40 所示，弹出“新建”对话框，选中“表”单选按钮，单击“确定”按钮，弹出“创建”对话框。



图 2-40 “数据库设计器”工具栏中的“新建表”按钮

选择“学生成绩管理信息系统”项目所在文件夹中的数据保存路径，输入表名“SelCourse.dbf”，单击“保存”按钮，打开表设计器，如图 2-41 所示。

新建数据库表还可以通过打开项目管理器，在“数据”选项卡中完成。

3. 创建数据表

按照任务描述创建数据库表 SelCourse.dbf 的结构。以“学号”字段为例，创建过程如下。

在“字段”选项卡中设置“名称”为“学号”，在“类型”下拉列表中选择“字符型”

选项，宽度设置为 6。

在“显示”选项组中，在“格式”文本框中输入格式码“T”，“输入掩码”文本框中输入格式掩码“999999”，“标题”文本框中输入字段，显示标题为“学生学号”。

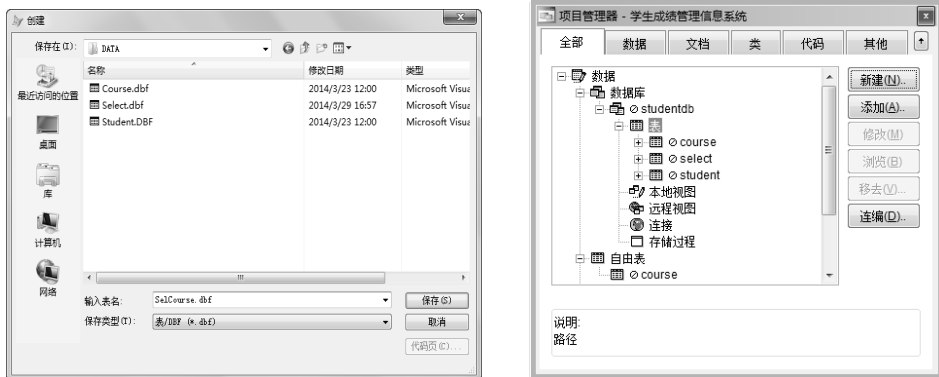


图 2-41 选择保存路径

在“字段有效性”选项组中，在“默认值”文本框中设置表达式 $\text{SubStr}(\text{Str}(\text{Year}(\text{Date()}), 4, 3, 2) + "9999")$ ，“学号”字段值由入学年份的末两位和四位流水号组成，如图 2-42 所示。

在“字段”选项卡中依次录入其他字段及其属性，如图 2-43 所示。

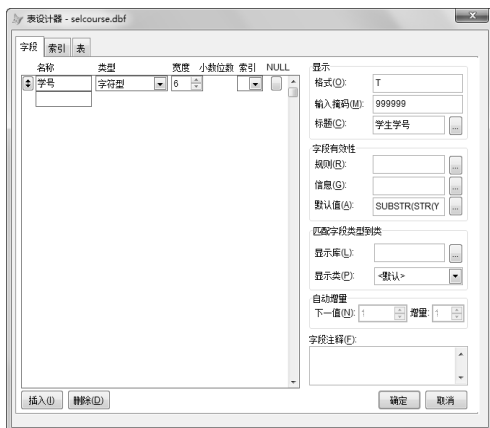


图 2-42 “学号”字段有效性设置结果



图 2-43 输入所有字段及属性

选择“表”选项卡，输入记录有效性规则，如图 2-44 所示。

单击“确定”按钮，完成数据库表 SelCourse.dbf 的创建。

本任务中字段有效性规则和记录有效性规则设置需要在本单元项目三的任务一“输入和追加表数据”一节中得到具体验证。



问题探究

1. 格式码字符和输入掩码字符在功能上有何区别？

格式码是对整个字段在显示输出格式上的一种约束规则。例如，限定整个字段内容均为字符，限定整个数值字段的表示方法为科学计数法等。



输入掩码字符是对字段内容在录入时的约束规则，每个掩码字符代表一个字符位或数字位。例如，要求录入 4 位数字字符位字段的掩码是“9999”。

2. 字段有效性规则和记录有效性规则的区别是什么？

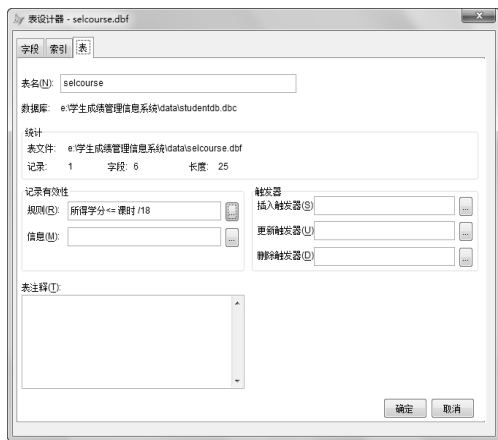


图 2-44 输入数据表的记录有效性表达式

字段有效性规则和记录有效性规则都是由表达式来约定的。字段有效性规则是在字段内容新增或发生改变时，进行规则检查。若不满足规则表达式，则该字段内容不允许通过。

记录有效性规则是在某条记录新增或发生改变时，自动进行规则检查。若该记录内容不满足记录有效性规则定义的表达式条件，则该记录内容不允许通过，一般记录有效性规则中的表达式包含多个字段值，表达了多个字段之间相互约束的关系。

任务四 修改数据库

任务目标

能够熟练修改数据库。

任务描述

已创建好的数据库在使用时会经常根据需求进行修改，修改内容如下。

在数据库中新建数据表。

在数据库中新建自由表。

移去或删除数据库中的表。

本任务对“学生成绩管理信息系统”数据库 StudentDB.dbc 进行修改。



实施步骤

有以下两种方法修改数据库。

1. 使用项目管理器修改数据库

选择“文件” 打开命令或单击常用工具栏中的“打开”按钮，在弹出的“打开”对话框

中选择“学生成绩管理信息系统.pjx”项目文件。

在项目管理器中选择数据库 StudentDB.dbc，单击“新建”按钮，可多次新建数据库表；单击“添加”按钮，可多次添加已有自由表；单击“修改”按钮，可打开数据库设计器进行修改；单击“移去”按钮，可以移去或删除数据库所包含的数据表；单击“关闭”按钮，关闭 StudentDB.dbc 数据库，如图 2-45 所示。



图 2-45 在项目管理器中操作数据库

2. 使用命令修改数据库

命令格式：

MODIFY DATABASE [带完整路径的数据库文件名]

功能：打开数据库设计器，修改当前打开的数据库文件。

执行如下命令：

MODIFY DATABASE StudentDB

数据库的修改是在数据库设计器中完成的。该命令执行后打开数据库设计器，便可以对当前数据库进行新建、添加、移去或删除数据对象的操作。

说明：该命令中的数据库文件名可以不带路径，但执行该命令的前提是系统能够在默认目录中找到该文件。



项目小结

根据学生成绩管理信息系统需求分析，本项目创建了数据库文件 StudentDB.dbc，并通过向该数据库中添加学生自然情况表 Student.dbf 和课程情况表 Course.dbf 及新建学生选修数据库表 SelCourse.dbf、修改数据库等内容，介绍了数据库创建与维护的方法和过程。



上机实习

1. 创建“图书管理信息系统”项目文件“图书管理系统.pjx”，并在该项目中新建图书管理数据库 BookDB.dbc。

2. 在图书管理数据库 BookDB.dbc 中添加图书自由表 Book.dbf、读者自由表 Loander.dbf、读者借书自由表 Loand.dbf。

3. 在图书管理数据库 BookDB.dbc 中新建读者借书数据库表 Loand_New.dbf，该表字段及其属性如表 2-13 所示。

表 2-13 读者借书数据库表 Loand_New.dbf 的表结构

字段名称	字段类型	字段宽度	小数位数	索引	NULL	显示	字段有效性	记录有效性
借书编号	整型	4				标题：借书序号	规则：借书编号>=0	借书时间<=还书时间
读者编号	整型	4					规则：读者编号>=0	
图书编号	整型	4					规则：图书编号>=0	
借书时间	日期型	8						



续表

字段名称	字段类型	字段宽度	小数位数	索引	NULL	显示	字段有效性	记录有效性
还书时间	日期型	8						
续借	逻辑型	1				标题：是否续借	默认值：.F.	
滞纳金	货币型	8				格式：\$	规则：滞纳金 ≥ 1 or 滞纳金=0 信息：“推迟还书需缴纳滞纳金，且滞纳金最少1元，没有上限。”	

4. 在“命令”对话框中完成如下操作。

在用户磁盘上新建“图书管理信息系统”文件夹。进入 Visual FoxPro 系统，使用命令创建“图书管理信息系统”文件夹下的图书管理数据库文件 BookDB.dbc 和员工管理数据库文件 Employee.dbc。

设置当前数据库为 BookDB.dbc。

在当前数据库中新增图书自由表 Book.dbf、读者自由表 Loander.dbf、读者借书自由表 Loand.dbf (先将自由表拷贝到该文件夹下)。

关闭数据库文件 BookDB.dbc 和 Employee.dbc。

删除数据库文件 Employee.dbc。

修改数据库 BookDB.dbc。

移去数据库表 Loand.dbf。

关闭图书管理数据库文件 BookDB.dbc。



知识巩固

一、选择题

- 一个自由表的字段名最多可以包含 () 个字符。
A. 256 B. 128 C. 20 D. 10
- 一个数据库表的字段名最多可以包含 () 个字符。
A. 256 B. 128 C. 20 D. 10
- 下列选项中不可以包含在数据库里的对象是 ()。
A. 项目 B. 数据表 C. 视图 D. 存储过程
- 对数据库的操作不包括 ()。
A. 新增自由表 B. 新建数据库表 C. 删除数据库表 D. 删除自由表
- 下列说法中不正确的是 ()。
A. 在数据库中移去数据表实际上是删除该表在库中的路径指针
B. 字段的有效性规则是对某字段值的取值范围进行约束
C. 格式码是对字段格式的显示进行控制，输入掩码是控制输入该字段的数据格式
D. 无论自由表还是数据库表都可以设置显示格式和有效性规则
- 数据库在创建后产生的文件扩展名是 ()。

- A . .dbf B . .dbc C . .dbt D . .dcx
7. 不论使用哪种方法创建表, 都会打开 () 对话框进行设计。
A . 表向导 B . 命令 C . 表设计器 D . 文件选择
8. 在数据库中新增一个自由表的命令是 ()。
A . ADD TABLE [数据表名] B . CREATE TABLE [数据表名]
C . USE [数据表名] D . CLOSE TABLES
9. 关于项目、数据库、数据表之间的关系, 下列说法中正确的是 ()。
A . 数据库包含项目和数据表
B . 项目包含数据库, 数据库包含数据表
C . 当删除数据库时, 数据库中的数据表一起被删除
D . 同一个数据表可以同时包含在多个数据库中
10. 打开数据库的命令是 ()。
A . MODIFY DATABASE B . OPEN DATABASE
C . DELETE DATABASE D . CREATE DATABASE
11. 修改数据表结构的命令是 ()。
A . MODIFY DATABASE B . MODIFY TABLE
C . MODIFY STRUCTURE D . USE [数据表名]
12. 删除数据库的命令是 ()。
A . DELETE DATABASE B . OPEN DATABASE
C . CLOSE DATABASE D . DELETE TABLES
13. 关闭当前打开的数据表的命令是 ()。
A . DELETE DATABASE B . USE
C . REMOVE TABLES D . DELETE TABLES
14. 下面 () 环境设置命令执行后, 创建同名库文件时系统会弹出是否改写提示对话框。
A . SET TALK ON B . SET TALK OFF
C . SET SAFETY ON D . SET SAFETY OFF
15. 某字符型字段的显示格式设置为 " !T ", 那么当该字段值为 " Safety " 时, 该字段显示的值应该为 ()。
A . " Safety " B . " Safety " C . " SAFETY " D . " safety "

二、填空题

1. 数据库文件的扩展名是_____, Visual FoxPro 中用来创建、维护数据库的可视化工具是_____。
2. 创建数据库的命令是_____; 打开数据库的命令是_____; 修改数据库的命令是_____; 关闭数据库的命令是_____; 删除数据库的命令是_____; 在数据库中添加数据表的命令是_____; 在数据库中移去数据表的命令是_____。
3. 有效性规则的创建即生成_____, 在_____中构建。
4. _____可以代替字段名显示在浏览窗口或表格栏标头中。



5. 一个数据库文件可以包括_____、_____、存储过程、数据连接等对象。

三、简答题

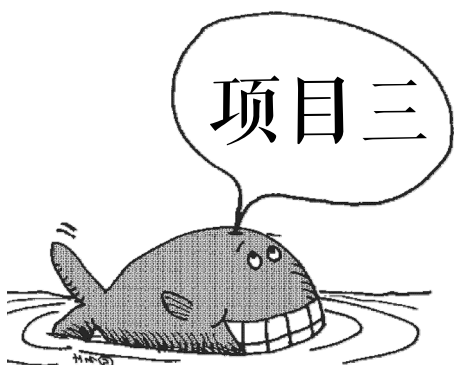
1. 自由表和数据库表的区别是什么？

2. 学生自然情况数据表中定义“学号”字段的数据类型为 C(6), 且这六位字符均为数字, 前两位为当前年份的末两位。例如, 学号“140001”中, “14”代表“2014”年份。请描述如何设置“学号”字段的显示格式和有效性规则？

3. 简述 3 种创建数据库的方法及步骤。

4. 简述在项目管理器中打开数据库设计器和表设计器的方法和步骤。

5. 简述项目、数据库和数据表之间的关系。



管理学生信息表数据

项目说明

在表结构建立起来后，就需要向表中输入数据了。对表数据的操作包括添加记录、修改和更新记录、删除和恢复记录、查找和排序记录及对不同表中的数据进行关联，Visual FoxPro 为每种操作都提供了交互式 and 命令两种方式。本项目通过 7 个任务分别完成这些操作，它们分别如下。

任务一：输入和追加表数据。

任务二：浏览表数据。

任务三：修改和更新表数据。

任务四：删除和恢复表数据。

任务五：学生信息排序。

任务六：检索学生数据。

任务七：多表的操作。

任务一 输入和追加表数据

任务目标

熟练使用菜单方式添加记录。

熟悉追加记录的命令并能熟练使用。

能够通过输入字段内容验证格式及有效性设置。

任务描述

在浏览窗口中打开学生自然情况表 Student.dbf，以追加方式输入如图 2-46 所示的表数据，然后依次打开课程情况表 Course.dbf 和学生选修数据库表 SelCourse.dbf，按图 2-47 及图 2-48



输入表数据。

学号	姓名	出生日期	性别	专业	籍贯	照片	父亲	母亲	联系电话	家庭住址
120101	杜嘉	03/03/98	男	计算机应用	辽宁	Gen	杜成	刘欣悦	13825000411	memo
120102	王晓明	11/07/98	男	计算机应用	黑龙江	Gen	王宏伟	张晓梅	15925418774	memo
120103	彭新信	05/22/99	女	计算机应用	辽宁	gen	彭程	岳江珊	13952415874	memo
120104	李丽	08/17/99	女	计算机应用	山东	Gen	李鸣居	王爱华	13322215858	memo
120105	顾晓雪	04/27/98	女	计算机应用	河北	Gen	顾春生	秦焕海	13644298745	memo
120201	金明	02/09/99	男	电子技术	吉林	Gen	金鑫	刘雪梅	15412457841	memo
120202	唐亦	10/10/98	女	电子技术	吉林	Gen	唐瑛	张爱生	15522255566	memo
120203	王敏敏	01/20/99	女	电子技术	黑龙江	Gen	王胜利	于鑫琴	15847415241	memo
120204	李璐	07/08/98	女	电子技术	山东	gen	李鑫进	徐梅	13624112352	memo
120205	杨建	06/30/99	男	电子技术	河北	gen	杨白河	高秋月	13645125878	memo

图 2-46 学生自然情况表数据

学生学号	课程编号	授课学期	所得学分	课时	期中成绩	期末成绩	平均成绩
120201	k008	1	8	144	89.0	76.0	81.2
120201	k009	4	4	72	40.0	40.0	40.0
120202	k007	2	4	72	56.0	45.0	49.4
120202	k008	1	8	144	60.0	60.0	60.0
120202	k009	4	4	72	93.0	76.0	82.8
120203	k005	3	4	72	87.0	93.0	90.6
120203	k008	1	8	144	50.0	30.0	38.0
120203	k010	3	4	72	0.0	0.0	70.0
120204	k005	3	4	72	75.0	65.0	69.0
120204	k007	2	4	72	92.0	84.0	87.2
120204	k009	4	4	72	73.0	45.0	56.2
120204	k010	3	4	72	0.0	0.0	50.0

图 2-47 课程情况表数据

课程编号	课程名称	学分	授课学期	参考学时	考试	推荐教材
k001	数据结构	4	4	68	T	gen
k002	计算机基础	2	2	34	F	gen
k003	计算机网络	4	4	68	T	gen
k004	数据库基础	6	3	102	T	gen
k005	专业英语	2	1	34	F	gen
k006	电工基础	4	3	60	T	gen
k007	数字电路	4	2	68	T	gen
k008	高等数学	8	1	160	T	gen
k009	传感器	4	4	68	T	gen
k010	微机原理	4	3	68	F	gen

图 2-48 学生选修数据表数据



实施步骤

可以使用交互方式和命令方式输入或追加表数据。

1. 使用交互方式输入表数据

(1) 创建好数据表结构后, 系统会提示用户“是否立即输入表数据?”, 单击“确定”按钮后, 系统自动打开浏览窗口进行表数据的输入。

在数据表关闭的情况下有 3 种方法打开浏览窗口。

通过在项目管理器中打开“学生成绩管理信息系统”, 选择“数据”选项卡, 展开数据库 StudentDB, 在该数据库所包含的表选择“Student”选项, 然后单击“浏览”按钮, 打开如图 2-46 所示的数据表浏览窗口, 操作过程如图 2-49 所示。

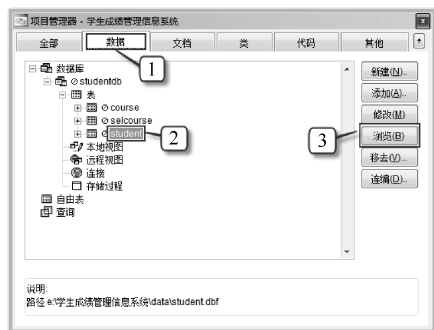


图 2-49 在项目管理器中浏览数据库表的方法

通过在数据库设计器中打开数据库 StudentDB, 右击数据表 Student, 在弹出的快捷菜单中选择“浏览”命令, 如图 2-50 所示。

通过在“命令”对话框中输入浏览命令“BROWSE”打开浏览窗口, 该命令在任务二中会详细介绍使用方法。

(2) 如果是第一次输入数据, 这时浏览器窗口只显示字段名一行, 没有记录, 且光标在表格中任何位置单击都不能直接输入或编辑数据, 数据表处于浏览模式, 如图 2-51 所示。



图 2-50 在数据库设计器中浏览数据库表的方法



图 2-51 数据表在浏览模式下的状态

(3) 选择“显示”“追加方式”命令后，在浏览窗口的最后一条记录后自动追加一条空记录，且记录内容变成可编辑状态，可以增加或修改字段内容，如图 2-52 所示。



图 2-52 数据表追加方式的设置

(4) 选择“显示”“编辑”命令后，浏览窗口中的记录变成编辑模式，“编辑”模式下，记录中的各字段呈纵向排列，如图 2-53 和图 2-54 所示。



图 2-53 数据表“编辑”模式的设置

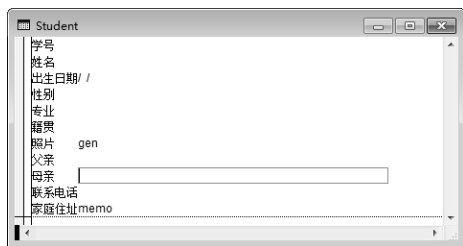


图 2-54 数据表在“编辑”模式下的状态

(5) 按照图 2-46 输入数据，具体输入方法如下。

输入字符型、数值型、逻辑型、日期型数据时，先单击需要修改的数据，便可进入编辑状态。

对备注型或通用型字段的输入方法：双击该字段内“memo”或“gen”标记，即可以编辑该字段。

例如，双击当前记录“照片”字段中的“gen”标记，进入照片字段编辑窗口。然后选择“编辑”→“插入对象”命令，如图 2-55 所示，弹出“插入对象”对话框，如图 2-56 所示，选中“由文件创建”单选按钮，单击“浏览”按钮选择已有照片文件，然后单击“确定”按钮。添加内容后的通用型和备注型字段的“gen”或“memo”标记自动转变成“Gen”或“Memo”形式。已插入对象的通用型字段的浏览结果如图 2-57 所示。

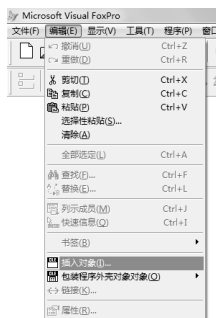


图 2-55 “插入对象”命令



图 2-56 “插入对象”对话框



图 2-57 浏览备注字段

(6) 选择“表” “追加新记录”命令，在浏览窗口最后一条记录后新增一条新记录。

(7) 选择“表” “追加记录”命令，弹出“追加来源”对话框，可以从其他数据表中将符合要求（即字段名和字段类型相同的字段）的记录添加到当前数据表的尾部，如图 2-58 所示。

(8) 按照图 2-47 和图 2-48 依次输入课程情况表 Course.dbf 和学生选修数据表 SelCourse.dbf 的数据。

2. 使用命令方式追加记录

命令格式：

APPEND [BLANK]

功能：在当前打开的数据表末尾追加记录。

参数：

[BLANK]：可选项。追加一条空记录。若省略该选项，则进入追加方式，用户可以追加多条记录。

在“命令”对话框中输入如下命令，完成追加多条记录的操作。

```
OPEN DATABASE StudentDB
```

&& 数据库 StudentDB 存放在默认路径下

```
USE Student
```

```
APPEND
```

&& 逐条追加多条记录

```
USE
```

```
CLOSE DATABASE
```

命令格式：

APPEND FROM <表文件名> [FIELDS 字段列表] [FOR 条件表达式]

功能：将指定表文件中的记录追加到当前表中，且只追加同名字段的数据。

参数：

● <表文件名>：必选项。带路径的源表文件名。

● [FIELDS 字段列表]：可选项。将源表中指定字段列表的值追加到当前表。若省略，则追加源表中与当前表的同名字段。

● [FOR 条件表达式]：可选项。将源表中符合条件的记录追加到当前表。若省略，则追加源表中所有记录。

● 在“命令”对话框中输入如下命令，完成从已有数据表中追加数据到当前表的操作。

```
OPEN DATABASE StudentDB
```

```
USE Student
```

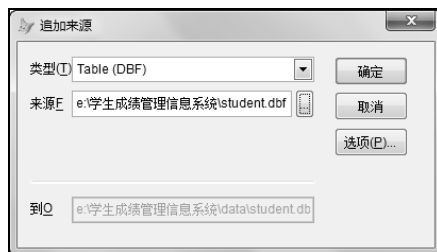


图 2-58 “追加来源”对话框

```

COPY STRU TO Student_1      && 将表结构复制到新表
USE                          &&关闭 Student
USE Student_1
APPE FROM Student FIELDS 学号,姓名,专业 FOR 专业=" 计算机应用 "
&& 向新表追加计算机应用专业的学生记录,且只追加学号、姓名和专业字段的值
USE                          &&关闭 Student_1
CLOSE DATABASE

```



问题探究

1. 在输入学生选修数据表 SelCourse 的数据内容时,字段值在显示格式及域值的选择上是否有限制?

以“授课学期”字段的输入为例,在表结构设计时该字段的属性如表 2-14 所示。显示格式和输入掩码限定了该字段只能输入一位数字字符,字段有效性规则限定了这位数值只能是“1”~“6”之间的数字,当该字段值的输入超出域值范围时,提示错误信息,如图 2-59 所示;当该条记录内容输入完毕,当前光标指针离开该记录时,系统按照记录有效性规则自动进行检测,如果“所得学分”与“课时”这两个字段值之间不能满足记录有效性规则,则提示错误信息,如图 2-60 所示。

表 2-14 学生选修数据表 SelCourse 的字段格式及有效性设置

字段名称	字段类型	字段宽度	小数位数	索引	NULL	显示	字段有效性	记录有效性
授课学期	字符型	1				格式:T 输入掩码:9	规则:授课学期>="1" and 授课学期<="6" 信息:"授课学期为"1"、"2"、..."6",分别代表6个学期。" 默认值:"1"	所得学分<=课时/18



图 2-59 字段有效性设置的验证及错误提示



图 2-60 记录有效性设置的验证及错误提示

任务二 浏览表数据

任务目标

能够使用交互方式浏览数据表。



能够使用浏览命令浏览数据。

任务描述

在浏览器中打开学生自然情况表 Student.dbf, 浏览记录; 改变字段的显示宽度和显示顺序, 定制浏览窗口; 根据条件定制显示的记录; 根据要求定制显示的字段。



实施步骤

1. 使用菜单, 交互式设置浏览数据表的格式和内容

在项目管理器中打开“学生成绩管理信息系统”项目文件, 选择数据库表 Student.dbf, 单击“浏览”按钮, 打开浏览窗口。或者在数据库管理器中打开数据库 StudentDB.dbc, 右击数据库表 Student.dbf, 在弹出的快捷菜单中选择“浏览”命令。

改变行高和列宽。浏览窗口中的表数据是以行和列的形式组织的表格。

将光标放置在第一条记录与第二条记录相交线的最左侧, 当光标变成十字形时, 按住鼠标左键, 这时记录间的相交线突出显示, 上下拖动鼠标指针可以改变记录显示的行高。Visual FoxPro 只能改变第一条记录的行高, 其他行与第一行高度相同。

将光标放置在浏览窗口标题栏字段名之间的交界线上, 当出现十字形光标时, 按住鼠标左键左右拖动, 可以任意改变字段宽度。或者选择“表” “调整字段大小”命令, 然后按住左、右箭头调整字段宽度, 按 Enter 键, 完成设置。

改变字段显示顺序。将光标放置在浏览窗口中需要移动的字段标题的上面, 按住鼠标左键拖动此字段到需要的位置, 调整字段的显示顺序。或者选择“表” “移动字段”命令, 然后按住左、右箭头拖动此字段到需要的位置。

拆分浏览窗口。将光标放置到浏览窗口左下角的拆分条(黑色小方块)上, 光标变成左右两个箭头, 按住鼠标左键并向右拖动至合适的位置放开, 即可将浏览窗口拆分成两部分。如果将拆分条拖到最左边, 这时可以将浏览窗口恢复成原来的一个窗口。

浏览窗口的两个子窗口是相互链接的, 它们显示相同数据表的内容, 在一个窗口中移动记录指针, 另一个窗口中的记录指针也会相应改动, 以保持两个子窗口中看到的是同一条记录。

定制显示的记录, 在浏览窗口显示“计算机应用”专业的学生记录。

选择“表” “属性”命令, 弹出“工作区属性”对话框。在“数据过滤器”文本框中设置筛选记录的条件, 如图 2-61 所示, 单击文本框右侧的按钮弹出“表达式生成器”对话框, 在过滤表达式文本框中输入过滤条件: Student. 专业=“计算机应用”, 如图 2-62 所示, 单击“确定”按钮后, 在浏览窗口中只显示定制后符合条件的记录, 如图 2-63 所示。

定制显示的字段, 在浏览窗口中选择显示所有记录的学号、姓名、性别、专业、联系电话字段。

选择“表” “属性”命令, 弹出“工作区属性”对话框。在“允许访问”选项组中选中“字段筛选指定的字段”单选按钮, 如图 2-64 所示, 然后单击“字段筛选”按钮, 弹出“字段选择器”对话框, 将学号、姓名、性别、专业、联系电话字段移动到“选定字段”列表框中, 单击“确定”按钮后, 在浏览窗口中就只显示定制后符合条件的字段了, 如图 2-65 所示。



图 2-61 使用数据过滤器定制显示的记录

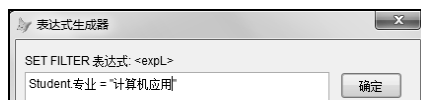


图 2-62 在表达式生成器中输入记录过滤条件

学号	姓名	出生日期	性别	专业	籍贯	照片	父亲	母亲	联系电话	家庭住址
120101	杜嘉	03/03/98	男	计算机应用	辽宁	Gen	杜成	刘欣悦	13825000411	memo
120102	王晓明	11/07/98	男	计算机应用	黑龙江	Gen	王宏伟	张晓梅	15925418774	memo
120103	彭新信	05/22/99	女	计算机应用	辽宁	gen	彭程	岳江珊	13952415874	memo
120104	李丽	08/17/99	女	计算机应用	山东	Gen	李鸣居	王爱华	13322215858	memo
120105	顾晓雪	04/27/98	女	计算机应用	河北	Gen	顾春生	秦焕海	13644298745	memo

图 2-63 数据过滤后的显示结果

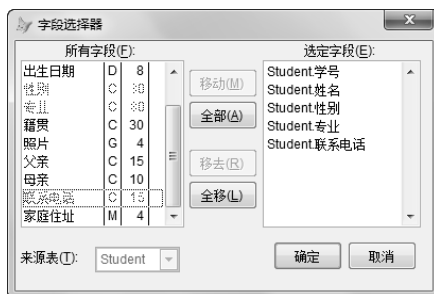


图 2-64 在“工作区属性”对话框中设置显示指定字段 图 2-65 在“字段选择器”对话框中选择显示的字段

2. 使用命令定制浏览数据的内容

命令格式：

BROWSE [FIELDS 字段列表] [FOR 条件表达式]

功能：打开浏览窗口浏览或编辑当前数据表的数据。

参数：

- [FIELDS 字段列表]：可选项。浏览字段列表中的数据。
- [FOR 条件表达式]：可选项。浏览条件表达式定义的记录。

使用命令浏览并编辑数据表时，首先应打开该表。在“命令”对话框中输入如下命令：

```
OPEN DATABASE StudentDB && 打开存放在默认路径下的数据库 StudentDB
USE Student
BROWSE FIELDS 学号,姓名,性别,专业,联系电话
BROWSE FOR 专业="计算机应用"
```



```
BROWSE FIELDS 学号,姓名,性别,专业,联系电话 FOR 专业=" 计算机应用 "
USE
CLOSE DATABASE
```

任务三 修改和更新表数据

任务目标

能对指定记录进行修改。

能使用菜单和命令方式更新一条记录或成批更新表数据。

相关知识

一个数据表可以存放多达十亿条记录，为了便于管理这些记录，Visual FoxPro 提供了一个记录指针。当打开表文件时，记录指针指向第一条记录。随着命令的执行，记录指针会发生移动。数据表第一条记录是首记录，记为 TOP；最后一条记录是尾记录，记为 BOTTOM。在第一条记录之前有一个文件起始标志，称为文件首 BOF (Beginning of File)；在最后一记录的后面有一个文件尾标识 EOF (End of File)。记录指针指向的记录是当前记录，对表文件的操作均是对当前记录进行的。

改变当前记录的操作可以通过菜单或命令进行。Visual FoxPro 提供了操作记录指针的菜单项，如图 2-66 所示，可以将当前指针定位到第一条记录、最后一条记录、上移一条记录、下移一条记录、移动到指定记录号、定位选定记录的范围。在数据表中记录指针的标记如图 2-67 所示。

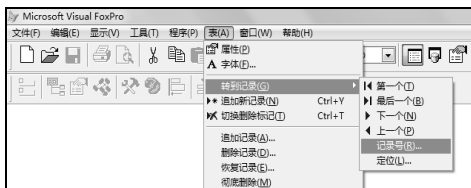


图 2-66 移动记录指针的菜单项



图 2-67 数据表中的记录指针标记

任务描述

在浏览器中打开学生选修数据表 Selcourse，如图 2-68 所示。按照图 2-69 的要求编辑修改字段值。

学生学号	课程编号	授课学期	所得学分	课时	期中成绩	期末成绩	平均成绩
120201	k008	1	8	144	89.0	76.0	81.2
120201	k009	4	4	72	40.0	40.0	40.0
120202	k007	2	4	72	56.0	45.0	49.4
120202	k008	1	8	144	60.0	60.0	60.0
120202	k009	4	4	72	93.0	76.0	82.8
120203	k006	3	4	72	87.0	93.0	90.6
120203	k008	1	8	144	50.0	30.0	38.0
120203	k010	3	4	72	0.0	0.0	70.0
120204	k006	3	4	72	75.0	65.0	69.0
120204	k007	2	4	72	92.0	84.0	87.2
120204	k009	4	4	72	73.0	45.0	56.2
120204	k010	3	4	72	0.0	0.0	50.0

图 2-68 学生选修数据表 Selcourse 的原始数据

记录号	修改字段	原值	修改后值
8	期中成绩	0.0	60
8	期末成绩	0.0	76
12	期中成绩	0.0	77
12	期末成绩	0.0	82

图 2-69 修改数据

更新所有记录的平均成绩字段值：平均成绩=期中成绩×40%+期末成绩×60%。



实施步骤

1. 使用菜单修改、更新表数据

在项目管理器或数据库管理器中打开学生成绩管理信息系统数据库 StudentDB.dbc，右击学生选修数据表 Selcourse，在弹出的快捷菜单中选择“浏览”命令，打开浏览窗口。

选择“表”“转到记录”“记录号”命令，弹出“转到记录”对话框，如图 2-70 所示，输入记录号 8，单击“确定”按钮。这时记录指针指向第 8 条记录，该行呈突出显示状态。

将光标移动到该条记录“期中成绩”字段内，按图 2-69 要求修改字段值，再修改“期末成绩”字段值。

选择“表”“转到记录”“记录号”命令，弹出“转到记录”对话框将当前记录号定位到 12，按图 2-69 的要求进行字段值的修改，修改后的结果如图 2-71 所示。



图 2-70 “转到记录”对话框

学生学号	课程编号	授课学期	所得学分	课时	期中成绩	期末成绩	平均成绩
120202	k008	1	8	144	60.0	60.0	60.0
120202	k009	4	4	72	93.0	76.0	82.8
120203	k006	3	4	72	87.0	93.0	90.6
120203	k008	1	8	144	50.0	30.0	38.0
120203	k010	3	4	72	60.0	76.0	70.0
120204	k006	3	4	72	75.0	65.0	69.0
120204	k007	2	4	72	92.0	84.0	87.2
120204	k009	4	4	72	73.0	45.0	56.2
120204	k010	3	4	72	77.0	82.0	50.0
120205	k008	1	8	144	90.0	70.0	78.0

图 2-71 按要求修改后的结果

更新“平均成绩”字段值。选择“表”“替换字段”命令，弹出“替换字段”对话框。在“字段”下拉列表框中选择“总成绩”字段（说明：“平均成绩”是该字段的显示标题），在“替换”文本框中单击右侧按钮，弹出“表达式生成器”对话框，输入如下表达式：

SelCourse.期中成绩*0.4 + SelCourse.期末成绩*0.6

在“替换条件”选项组中的“范围”下拉列表框中选择“ALL”选项，然后单击“替换”按钮，设置如图 2-72 所示。替换结果如图 2-73 所示，和替换前相比，所有记录的“平均成绩”字段全部进行了更新，在第 8 条和第 12 条记录上可以看出结果的变化。



图 2-72 “替换字段”对话框

学生学号	课程编号	授课学期	所得学分	课时	期中成绩	期末成绩	平均成绩
120202	k008	1	8	144	60.0	60.0	60.0
120202	k009	4	4	72	93.0	76.0	82.8
120203	k006	3	4	72	87.0	93.0	90.6
120203	k008	1	8	144	50.0	30.0	38.0
120203	k010	3	4	72	60.0	76.0	70.0
120204	k006	3	4	72	75.0	65.0	69.0
120204	k007	2	4	72	92.0	84.0	87.2
120204	k009	4	4	72	73.0	45.0	56.2
120204	k010	3	4	72	77.0	82.0	80.0
120205	k008	1	8	144	90.0	70.0	78.0

图 2-73 替换“总成绩”字段后的结果



2. 使用命令修改、更新表数据

命令格式：

```
REPLACE [范围] <字段名 1> WITH <表达式 1>[, <字段名 2> WITH <表达式 2>...] [FOR 条件表达式]
```

功能：成批替换字段值。

参数：

- <字段名 1> WITH <表达式 1>[, <字段名 2> WITH <表达式 2>.....]：必选项。用表达式 1 的值替换字段 1 的内容，用表达式 2 的值替换字段 2.....

- [范围]：可选项。指定要替换的记录范围。省略时只替换当前记录的值。只有在范围内的记录才能被替换。

可以被用为范围的关键字如下。

ALL：所有记录。

NEXT 记录数：从当前记录开始，包含当前记录在内，向记录尾方向数 N 条记录的范围。

RECORD n RecordNumber：指定记录号的记录。

REST：从当前记录开始到记录尾的所有记录。

- [FOR 条件表达式]：可选项。满足条件表达式的记录才能被替换。

在“命令”对话框中输入如下命令：

```
OPEN DATABASE StudentDB      && 打开存放在默认路径下的数据库
USE SelCourse
GO 8
BROWSE
    &&打开浏览窗口，记录指针定位在第 8 条记录，按照图 2-69 修改第 8 条记录
GO 12
BROWSE
    &&打开浏览窗口，记录指针定位在第 12 条记录，按照图 2-69 修改第 12 条记录
REPLACE ALL 总成绩 WITH 期中成绩*0.4 + 期末成绩*0.6
    &&将所有记录的总成绩字段值按公式替换
BROWSE
    &&浏览替换结果
USE
CLOSE DATABASE
```

其中，GO 命令为记录指针移动命令，具体命令格式如下。

```
GO TOP|BOTTOM|<记录号>
```

功能：移动当前表的记录指针。

参数：

- TOP：移动当前表记录指针到第一条记录。

- BOTTOM：移动当前表记录指针到最后一条记录。

- <记录号>：移动当前表记录指针到指定记录号的记录。



问题探究

1. 如何根据相对记录号进行当前记录指针的跳转？

命令格式：

```
SKIP [记录数]
```

功能：以当前记录为基准，移动当前表的记录指针。

参数：

[记录数]: 可选项。当记录数为正数时, 记录指针从当前位置向文件尾方向移动; 当记录数为负值时, 记录指针从当前位置向文件头方向移动。若该项被省略, 则记录指针从当前位置向文件尾移动一条记录。

例如:

```
SKIP 5          &&记录指针从当前表当前位置向文件尾移动 5 条记录
SKIP -8         &&记录指针从当前表当前位置向文件头移动 8 条记录
SKIP           &&记录指针从当前表当前位置向文件尾移动 1 条记录
```

2. 如何在 REPLACE 语句中使用范围参数更新第 8 条记录的总成绩字段值?

执行如下命令:

```
REPLACE RECORD 8 总成绩 WITH 期中成绩*0.4 + 期末成绩*0.6
&&使用范围参数确定替换的记录
```

该命令等同于如下两条命令:

```
GO 8
REPLACE 总成绩 WITH 期中成绩*0.4 + 期末成绩*0.6
&&将当前记录的总成绩字段值按公式替换
```

任务四 删除和恢复表数据

任务目标

- 能使用菜单和命令方式逻辑删除记录。
- 能使用菜单和命令方式恢复被逻辑删除的记录。
- 能使用菜单和命令方式物理删除记录。

相关知识

没有作用的记录需要被删除。Visual FoxPro 提供了删除记录的命令和菜单。在主菜单项“表”的下拉菜单中, 有“删除记录”、“彻底删除”和“恢复记录”3 个功能, 它们体现了逻辑删除和物理删除的概念。“删除记录”是逻辑删除该记录, 即在数据表文件中对该记录做了一个删除标记, 但记录并未真正从磁盘上抹去。“彻底删除”是将做了删除标记的记录从物理磁盘上彻底删除。除去删除标记即可恢复被逻辑删除的记录, 而数据被物理删除后不能被恢复。

任务描述

在浏览器中打开学生自然情况表 Student.dbf, 逻辑删除多余记录; 恢复错误删除的记录; 物理删除多余记录; 清空全部记录。



实施步骤

1. 使用菜单删除和恢复数据

(1) 打开数据库表 Student.dbf

在项目管理器中打开学生成绩管理信息系统项目文件, 选择数据库表 Student.dbf, 单击“浏



览”按钮, 打开浏览窗口。或者在数据库管理器中打开数据库 StudentDB.dbc, 右击数据库表 Student.dbf, 在弹出的快捷菜单中选择“浏览”命令。

(2) 逻辑删除第 3 条、第 10 条记录

选择“表”“删除记录”命令, 弹出“删除”对话框, 在“范围”下拉列表中选择“Record”选择, 在微调框中输入“3”, 单击“删除”按钮后 (如图 2-74 所示), 第 3 条记录左侧删除标记位被黑色填充, 表明该记录已被逻辑删除, 如图 2-75 所示。

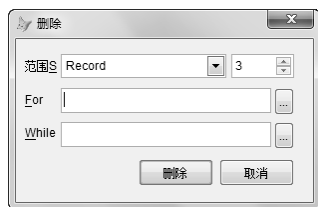


图 2-74 “删除”对话框

学号	姓名	出生日期	性别	专业
120101	杜嘉	03/03/98	男	计算机应用
120102	王晓明	11/07/98	男	计算机应用
120103	彭新信	05/22/99	女	计算机应用
120104	李丽	08/17/99	女	计算机应用
120105	逻辑删除标记	7/98	女	计算机应用
120106	李明	02/09/99	男	电子技术

图 2-75 当前记录的删除标记

在浏览窗口中直接单击第 10 条记录, 可以移动当前记录指针到该记录, 然后在该行左侧删除标记处单击, 也可以逻辑删除该记录。

(3) 恢复第 3 条记录

在浏览窗口中直接单击第 3 条记录, 使当前记录指针指向该记录, 然后单击该条记录左侧的黑色删除标记, 去除删除标记后当前记录被恢复。

(4) 物理删除第 10 条记录

选择“表”→“彻底删除”命令, 弹出如图 2-76 所示的提示对话框, 单击“是”按钮后, 系统将彻底从磁盘删除所有已被逻辑删除的记录。物理删除后, 这些记录将不能恢复。



图 2-76 提示对话框

2. 使用命令删除和恢复数据

命令格式:

```
DELETE [范围] [FOR 条件表达式]
```

功能: 逻辑删除当前表中的指定记录。

参数:

- [范围]: 可选项。指定要删除的记录范围。省略时只删除当前记录。只有在范围内的记录才能被删除。可以用于范围使用的关键字同 REPLACE 命令。

- [FOR 条件表达式]: 可选项。满足条件表达式的记录才能被逻辑删除。

命令格式:

```
RECALL [范围] [FOR 条件表达式]。
```

功能: 恢复当前表中在指定范围内符合条件表达式的记录中已被逻辑删除的记录为正常记录。

参数: 同 DELETE 命令。

命令格式:

```
PACK
```

功能: 物理删除当前表中已被逻辑删除的记录。

命令格式:

```
ZAP
```

功能：物理删除当前表中所有记录。

在“命令”对话框中输入如下命令：

OPEN DATABASE StudentDB	
USE Student	
DELETE 3	&&逻辑删除第 3 条记录
RECALL 3	&&恢复第 3 条记录
DELETE10	&&逻辑删除第 10 条记录
PACK	&&物理删除所有已被逻辑删除的记录
ZAP	&&物理删除所有记录，这个命令要谨慎使用
USE	
CLOSE DATABASE	

任务五 学生信息排序

任务目标

- 理解并掌握索引的分类和索引文件的分类。
- 能够建立不同类型的索引。
- 能够打开索引、确定当前索引并使用索引排序。

相关知识

记录被输入后在磁盘上是按照输入顺序被保存在数据表中的，这个顺序称为物理顺序。索引是按照某种规则对数据表中的记录进行逻辑排序，它是在逻辑上按照索引关键字对记录进行的升序或降序排序。为表建立索引后，系统会自动生成索引文件，该文件记录了按照索引排序后的记录顺序。索引并不改变记录在存储时的物理顺序。一张数据表可以建立多个索引。

1. 索引的分类

Visual FoxPro 中的索引有主索引、候选索引、唯一索引和普通索引 4 种。

主索引：只能在数据库表中建立，自由表不能建立主索引。在指定的一个或几个字段组成的主索引表达式中，索引值绝对不允许有重复，即主索引应该能唯一标识一条记录。一张表只能创建一个主索引。创建主索引所使用的关键字段被称为主关键字。

候选索引：在数据表中，能够唯一标识一条记录的关键字可能有多，选出其中一个作为主索引，其余的可以作为候选索引，也就是说，候选索引的关键字也不允许有重复值。在数据库表和自由表中均可以建立多个候选索引。创建候选索引使用的关键字被称为候选关键字。

唯一索引：唯一索引允许索引关键字重复，但在生成的索引文件中，只保留第一次出现的重复数据而去掉以后出现的相同字段值。一张表允许建立多个唯一索引。

普通索引：普通索引也允许索引关键字重复，并且在生成的索引文件中，索引项也允许出现重复值。一张表可以建立多个普通索引。

2. 索引文件

索引文件的分类如图 2-77 所示。

单项索引文件：扩展名为“.idx”，只能容纳一个索引，且只有升序排列方式，只能使用命



令创建。

结构复合索引文件 :索引文件名与数据表同名,扩展名为“.cdx”。一个复合索引文件可以包含多个索引,不同索引使用不同的索引标识 TAG 来区别,它既可以是升序,也可以是降序。

非结构复合索引文件 :索引文件名与表文件名不同,扩展名也为“.cdx”。一个非结构复合索引文件也可以包含多个索引,升序或降序排列。非结构复合索引不能创建主索引,且只能使用命令创建。

单项索引和非结构复合索引的文件名是由用户定义的,在使用前,必须使用单独的命令打开索引文件后方能使用。由于结构复合索引文件与表文件同名,在打开或关闭表文件的同时,系统自动打开或关闭与该表文件同名的结构复合索引文件。

当前数据表同时只能按照一个索引进行排序,该索引称为主控索引。使用时应先设置主控索引,才能按顺序显示记录。

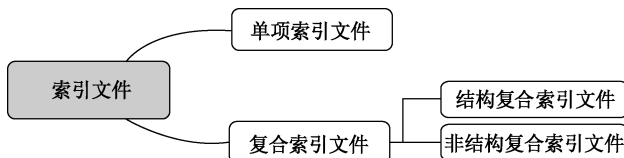


图 2-77 索引文件的分类

任务描述

对学生自然情况表 Student.dbf 建立索引。

建立基于“出生日期”字段的单项索引文件 StuBirth.idx。

建立基于“出生日期”字段的结构复合索引文件,索引标识为 BIRTH,按降序排列显示;在表文件的结构复合索引文件中,按“姓名+学号”建立索引,索引标识为 NUMNAME,按升序排列显示。

在表文件中,按“性别+专业”建立一个非结构复合索引文件 StuSx.cdx,索引标识为 SexS,按降序排列显示。



实施步骤

1. 使用交互方式建立索引

在项目管理器或数据库设计器中打开学生成绩管理数据库 StudentDB.dbc,选择学生自然情况表 Student.dbf 右击,在弹出的快捷菜单中选择“修改”命令,打开表设计器。

排序	名称	类型	表达式	筛选	排序
↑	学号	主索引	学号		Machine
↓	BIRTH	普通索引	出生日期		Machine
↑	NUMNAME	普通索引	学号+姓名		Machine

图 2-78 数据表 Student.dbf 中建立的索引

选择“索引”选项卡,在该选项卡中,已经存在一条以“学号”为索引字段的主索引。按照要求依次插入如图 2-78 所示的索引。

说明:

- 单项索引文件和非结构复合索引文件必须由命令创建。
- 常用的排序内容有 3 种: Machine——按机器码顺序排序; Pinyin——按拼音顺序排序; Stroke——按中文笔画顺序排序。
- 索引字段由当前表中的字段或由字段、函数、常数等组成的表达式构成。索引表达式可以是单一字段,也可以是多个字段的组合表达式。可以作为索引字段的类型是字符型、日期型、数值型和逻辑型。备注型和通用型字段不能作为索引字段。

2. 使用命令建立索引并按索引排序

(1) 建立索引

命令格式：

```
INDEXON<索引表达式>TO<单项索引文件>|TAG<索引标识名>[OF<复合索引文件名>]
[ASCENDING/DESCENDING] [UNIQUE|CANDIDATE]
```

功能：建立索引文件。该命令默认创建普通索引。

参数：

- <索引表达式>：必选项，由单个字段或多个字段组合而成。
- TO <单项索引文件>：创建一个单项索引文件，默认的扩展名是“.idx”。
- TAG <索引标识名> [OF <复合索引文件名>]：创建一个复合索引文件。[OF <复合索引文件名>]省略时，创建结构复合索引文件，索引文件与表文件同名，扩展名为“.cdx”。
- [ASCENDING/DESCENDING]：可选项，指定索引文件是升序或降序，默认时是升序。本选项只对复合索引文件有效。

- [UNIQUE|CANDIDATE]：唯一索引或候选索引。候选索引只对结构复合索引标识有效。

(2) 打开/关闭索引文件

使用索引文件必须先打开表文件，索引文件不能和表文件分开使用。打开索引文件的命令有以下两个。

用 USE 命令打开表文件的同时打开索引文件。命令格式：

```
USE<表文件名> [ INDEX <索引文件名>/ [ORDER TAG <索引标识> [OF <复合索引文件名>]
[ASCENDING/DESCENDING] ]]
```

功能：打开表文件同时打开索引文件并建立主控索引。

参数：

- INDEX <索引文件名>：打开单项索引文件或非结构复合索引文件。
- [ORDER TAG <索引标识> [OF <复合索引文件名>]：设置某复合索引文件中的索引标识为主控索引。
- [ASCENDING/DESCENDING]：可选项。指定索引文件是升序或降序排列，默认时是升序。

单独打开索引文件。命令格式：

```
SET INDEX TO [<索引文件名>/ [ORDER TAG <索引标识> [OF <复合索引文件名>]
[ASCENDING/DESCENDING] ]
```

功能：打开或关闭索引文件。

参数：同上。不带附加参数的 SET INDEX TO 命令，将关闭在当前工作区中所有打开的索引文件。

设置主控索引

当有多个索引文件同时被打开或一个索引文件中包含多个索引项时，同一时刻只能按照一个索引进行排序，该项索引被称为主控索引。

命令格式：

```
SET ORDER TO [<单项索引文件名>/ [TAG] <索引标识> [OF <复合索引文件名>]
[ASCENDING/DESCENDING] ]
```

功能：确定某项索引为主控索引。

参数：同上。

删除索引

可以在表设计器中的“索引”选项卡中删除索引，也可以使用命令删除索引。



命令格式：

```
DELETE TAG ALL|<索引名>
```

功能：删除打开的结构复合索引文件的索引标识。

在“命令”对话框中输入如下命令：

```
OPEN DATABASE StudentDB
USE Student
INDEX ON 出生日期 TO StuBirth
                                &&建立单项索引文件 StuBirth.idx 并打开索引文件

BROWSE
INDEX ON 出生日期 TAG Birth DESC
                                &&建立结构复合索引文件，索引字段为出生日期

BROWSE
INDEX ON 学号+姓名 TAG NumName
                                &&在结构复合索引文件中，以学号+姓名为索引项建立索引，升序排列

BROWSE
INDEX ON 性别+专业 TAG SexS OF StuSx DESC
&&建立非结构复合索引文件 StuSx.cdx，索引项为性别+专业，索引标识 SexS，降序排列
BROWSE
USE
SET INDEX TO                    &&关闭索引文件
CLOSE DATABASE
```



问题探究

1. 当表文件和索引文件都被关闭时，如何再次使用某个索引？

索引文件必须和表文件同时使用，并且使用前需要打开索引文件，确定主控索引，然后才能按照主控索引字段显示和访问数据。

在表文件和索引文件都被关闭的情况下使用索引文件的命令如下。

```
OPEN DATABASE StudentDB
Use Student                    &&打开表文件，同时打开结构复合索引文件
SET INDEX TO StuSx.cdx
                                &&打开非结构复合索引文件

SET ORDER TO Birth            &&将结构复合索引文件中索引项 Birth 设置为主控索引
BROWSE
SET ORDER TO NumName
                                &&将结构索引文件中索引项 NumName 设置为主控索引

BROWSE
SET ORDER TO SexS OF StuSx
                                &&将非结构索引文件 StuSx.cdx 中索引项 SexS 设置为主控索引

BROWSE
SET INDEX TO                    &&关闭当前所有索引
USE
CLOSE DATABASE
```

2. 为何使用组合字段的索引？

组合字段的索引可以使记录排序更加有序。例如，使用“姓名+性别”组合字段建立的索引，在姓名出现重名时，为了区别重名记录，可以在索引项中再添加“性别”字段，若添加“性别”字段后还是不能区分，则可以考虑使用多个字段的组合建立索引项。

3. 如何使用索引检索学生数据？

为表文件建立索引后，当索引文件打开并设置了主控索引时，记录按照索引字段升序或降序排列，这样按照索引关键字段值进行检索或查询时，计算机的执行速度会大大提升，记录数越多时，查询效果越明显。

Visual FoxPro 提供了索引查询命令及检测查询结果的常用函数。

命令格式：

```
SEEK <表达式> [ORDER <单项索引文件名>/ [TAG] <索引标识>] [OF <复合索引文件名>]
[ASCENDING/DESCENDING] ]
```

功能：在打开的索引文件中查找主控索引关键字与<表达式>相匹配的第一条记录，并将指针定位在该记录上。

参数：在指定查找条件时同时指定当前主控索引。若全部缺省参数，索引文件的打开和设置主控索引的命令应在执行该命令前执行。

与 SEEK 命令配合使用的用来检测查找结果的常用函数有以下 3 个，如表 2-15 所示。

Found()，返回逻辑值。若返回值为“.T.”，则表明找到匹配记录。

EOF()，返回逻辑值。若返回值为“.T.”，则表明查找到文件尾。

RecNO()，返回查找结束后指针定位所在记录号。

表 2-15 常用表查询检测函数

SEEK 查询结果	Found()	EOF()	RecNO()
查询成功	.T.	.F.	匹配记录号
查询失败	.F.	.T.	表全部记录数+1

【例 2-1】在“命令”对话框中输入如下命名：

```
OPEN DATABASE StudentDB
USE Student
SET INDEX TO StuSx.CDX                                &&打开非结构复合索引文件
SET ORDER TO SexS
```

&&将非结构索引文件 StuSx.cdx 中索引项 SexS 设置为主控索引，索引关键字为“性别+专业”

```
BROWSE
SEEK "男计算机应用"
?Found()
?EOF()
?RecNO()
SET INDEX TO                                           &&关闭当前索引文件
USE
CLOSE DATABASE
```

运行结果：

```
.T., .F., 5
```

表明找到第 5 条记录为匹配项。

任务六 检索学生数据

任务目标

能够移动记录指针并按条件检索学生记录。

任务描述

在学生自然情况表 Student.dbf 中查找籍贯是山东的所有学生。



相关知识

检索数据可以使用索引来完成,也可以在没有建立索引的情况下检索指定数据。Visual FoxPro 提供了菜单和命令两种方法进行数据检索。



实施步骤

1. 使用菜单完成数据检索

在数据库设计器中打开 StudentDB.dbc, 选择数据表 Student.dbf。

选择 “表” “转到记录” “定位” 命令, 弹出 “定位记录” 对话框。

在 “For” 文本框中输入条件表达式 “student.籍贯='山东'”, 单击 “定位” 按钮, 如图 2-79 所示。

双击学生自然情况表 Student.dbf, 打开浏览窗口, 检查记录指针所指的当前记录是否符合检索条件。

再次执行定位查询, 发现光标还是定位在相同记录上。使用菜单检索只能查找满足条件的第一条记录, 如图 2-80 所示。

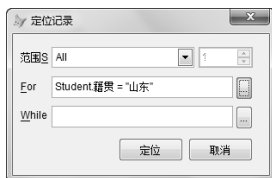


图 2-79 “定位记录”对话框

Student						
学号	姓名	出生日期	性别	专业	籍贯	
120202	唐莎	10/10/98	男	电子技术	吉林	
120205	杨建	06/30/99	男	电子技术	河北	
120101	杜嘉	03/03/98	男	计算机应用	辽宁	
120103	彭新信	05/22/99	男	计算机应用	辽宁	
120104	李丽	08/17/99	男	计算机应用	山东	
120105	顾晓雪	04/27/98	男	计算机应用	河北	
120201	金明	02/09/99	女	电子技术	吉林	
120203	王敏敏	01/20/99	女	电子技术	黑龙江	
120204	李璐	07/08/98	女	电子技术	山东	
120102	王晓明	11/07/98	女	计算机应用	黑龙江	

图 2-80 使用菜单检索只能定位第一条符合条件的记录

2. 使用命令检索

Visual FoxPro 提供了检索数据的一组命令。

命令格式：

LOCATE FOR <条件表达式>

功能：在当前表中查找满足条件的第一条记录, 若检索成功, 则将当前记录指针指向该记录, 否则指针移动到文件尾。

如果要检索满足条件的所有记录, 可以使用继续查找命令。

命令格式：

CONTINUE

功能：与 LOCATE 命令组合使用, 继续查找下一条符合条件的记录。若检索成功, 则将当前记录指针指向该记录, 否则指针移动到文件尾。

说明：

- 命令 CONTINUE 必须和 LOCATE 组合使用，它们只移动记录指针，不能浏览结果。
- 这两条命令的查找结果可以通过测试函数 Found()、EOF()、RecNO()检测。函数的不同查询结果返回值如表 2-15 所示。

在“命令”对话框中输入如下命令：

```

OPEN DATABASE StudentDB
USE Student
LOCATE FOR 籍贯="山东"
?Found( ),EOF( ),RecNO( )
CONTINUE
?Found( ),EOF( ),RecNO( )
USE
CLOSE DATABASE

```

&&查找满足条件的第一条记录
 &&输出 3 个检测函数值
 &&从当前记录开始向文件尾继续查找
 &&再次输出查找结果



问题探究

1. SEEK 命令和 LOCATE 命令的区别是什么？

SEEK 是索引查询命令，它必须在索引文件中通过查找索引字段得到查询结果，且必须是精确查询。LOCATE 命令不需要依赖索引文件，且查询表达式可以由任何字段组成的表达式，条件设定范围更宽。但是 SEEK 命令的查询效率更高。

2. 如何将所有符合条件的记录都查找出来？

在单元四中将会介绍结构化程序设计的 3 种结构，其中一种是循环结构。可以设计循环语句查询所有符合条件的记录，命令如下：

```

OPEN DATABASE StudentDB
USE Student
LOCATE FOR 籍贯="山东"
DO WHILE NOT EOF( )
&&若记录指针没有检索到文件尾，则执行 DO 和 ENDDO 中的语句，直到将文件检索完毕
?RecNO( )
CONTINUE
ENDDO
USE
CLOSE DATABASE

```

&&输出查找到的记录号
 &&移动记录指针，从当前记录开始向文件尾继续查找

任务七 多表的操作

任务目标

- 掌握多工作区的概念，熟练使用设置工作区的方法。
- 能够创建表间临时关系，并同时浏览多表数据。
- 能够创建表间永久关系，并设置参照完整性。

任务描述

同时打开学生自然情况表 Student.dbf、课程情况表 Course.dbf 及学生选修数据表 SelCourse.dbf，分别指定别名为 Stu、Cou、SelC。

以自然情况表为主表，创建自然情况表和选修数据表之间一对多的临时关系；以课程



情况表为主表, 创建课程情况表与选修数据表之间一对多的临时关系。浏览多表的数据, 其中包括字段学号、姓名、专业、课程编号、期中成绩、期末成绩、总成绩。

创建自然情况表与选修数据表之间一对多的永久关系; 创建课程情况表与选修数据表之间一对多的永久关系。创建如表 2-16 所示的参照完整性设置。

表 2-16 数据库表 Student.dbf 与 Course 之间参照完整性设置

父 表	子 表	更 新 规 则	删 除 规 则	插 入 规 则
Student	SelCourse	级联	限制	忽略
Course	SelCourse	级联	限制	忽略

相关知识

1. 工作区

数据表的打开都是在工作区中进行的, 在一个工作区中使用 USE 命令同时只能打开一张表。此时若想再使用 USE 命令在同一工作区中打开另一张表, 原来打开的表就会自动关闭。为了满足同时对多表进行操作, Visual FoxPro 提供了多工作区的操作方式。

当前工作区和当前表。在 Visual FoxPro 中最多可以有 32767 个工作区, 使用不同工作区可以同时处理不同的表。最后一个被选中的工作区称为“当前工作区”, 在当前工作区打开的表称为当前表。

工作区名。工作区名也称为工作区号 (1 ~ 32767), 在选择工作区时使用。系统刚启动时, 工作区 1 被自动选中。

工作区别名。每个工作区除了有一个工作区名外, 还可以有两个别名。一个是系统别名, 另一个是在该工作区中打开的表名或用户为打开的表定义的别名。

- 系统别名: 工作区 1 ~ 10 的系统别名分别用 A ~ J 单个字母来命名。工作区 11 ~ 32767 的系统别名分别是 W11 ~ W32767。

- 表名或用户定义的表别名: 若用户在某工作区打开表时未给表赋予另外的别名, 则该工作区上打开的表名即为其别名。

在工作区中打开表的命令格式:

```
USE <表名> [ALIAS <别名>] [IN <工作区号>]
```

功能: 在工作区中以别名打开表。

在工作区中关闭表的命令格式:

```
USE [IN <工作区号>] <别名>
```

功能: 关闭不同工作区数据表。

【例 2-2】

```
USE Student ALIAS Stu      &&在当前工作区中打开表, 别名为 Stu
USE Course IN 2            &&在工作区 2 中打开表, 别名即表名
USE SelCourse ALIAS SelC IN 3 &&在工作区 3 中打开表, 别名为 SelC
```

选择工作区的命令格式:

```
SELECT <工作区号>|<别名>
```

功能: 选择当前工作区, 用于对该工作区的表进行操作。

说明:

- 当工作区号是 0 时, 表示选定当前未使用的最小号工作区。
- 在表文件被打开后才可以使用别名选择工作区。

使用多工作区中的表数据。在当前工作区中可以直接使用表中的字段，若在当前工作区中使用其他工作区中的表字段，必须使用表别名进行标识，格式为：表别名.字段名或者表别名->字段名。

2. 建立表间临时关系，同时浏览多表数据

工作区为同时打开并操作多表创造了条件。Visual FoxPro 可以在当前表和其他打开的表之间建立一对一或一对多的关系。两个表建立关系后，当前表被称为主表，与之建立关系的表称为子表。主表和子表通过相同字段建立关联。若主表与子表的公共字段都是主索引或候选索引时，主表中的一条记录在子表中最多只有一条记录与之对应，那么两个表之间建立的是一对一关系。若主表的公共字段是主索引或候选索引，子表的公共字段为普通索引，主表中的一条记录在子表中有多条记录与之对应，那么两个表之间建立的是一对多关系。多对多的关系在表设计时最好分解成多张表，通过建立多个一对多的关系来解决。

操作时，临时关系可以通过“数据工作期”对话框来建立，也可以通过命令建立。在建立临时关系前，子表必须要先使用公共字段作为关键字建立索引。

临时关系建立后，子表中的记录指针跟随主表的记录指针一起移动，当选择主表的一条记录后，子表就会显示与主表索引关键字的值相同的所有匹配记录。在数据表关闭后，临时关系也取消。

(1) 创建一对一关系的命令

命令格式：

```
SET RELATION TO [关键字 INTO 别名]
```

功能：以当前表为主表，与指定别名的表在关键字段上建立一对一关系。

参数：

[关键字 INTO 别名]：关键字即为主表与子表建立关系的公共字段，别名为子表的别名。

参数缺省时，取消关系。

(2) 创建一对多关系的命令

命令格式：

```
SET SKIP TO <别名>
```

功能：以当前表为主表，与指定别名的子表在关键字段上建立一对多关系。

参数：

<别名>：必选项。子表的别名。

说明：在建立一对多关系之前必须先以该关键字建立一对一关系。

【例 2-3】

```
USE Student ALIAS Stu IN 1
USE Course IN 2
USE SelCourse ALIAS SelC IN 3
SELE 3
SET ORDER TO TAG 学号
```

&&此时，当前工作区仍为 1

&&设置主控索引为学号字段

&&以下命令建立关系后，在两个浏览窗口分别观察

关联表间数据的显示

```
SELE 1
SET RELATION TO 学号 INTO SelC
```

&&与别名为 SelC 的工作区表(子表)建立临时关系

&&浏览主表 Student 数据，窗口不要关闭

BROWSE

```
SELE 3
```



BROWSE

表当前记录改变时，子表浏览窗口中只显示与主

&&同时浏览子表 SelCourse 的数据。此时观察，当主

&&表学号一致的所有记录

&&执行以下命令前，首先关闭两个浏览窗口

```
SELE 1
```

```
BROWSE FIELDS 学号,姓名,专业,SelC.课程编号,SelC.期中成绩,SelC.期末成绩,SelC.总成绩
```

```
CLOSE TABLES ALL
```

3. 建立表间的永久关系，创建参照完整性

永久关系是数据库内表的一种关系，存储在数据库中，主要功能是使主表和子表在数据更新、删除、插入时保持一致。数据库表间永久关系是用来设置参照完整性信息的，它不能实现主表与子表间记录指针联动或同步移动。当存储有永久关系的数据库被打开时，永久关系即可使用。

永久关系可以使用菜单方式建立。

当主表和子表之间建立了永久关系后，主表和子表之间就以公共关键字段建立了关联。当主表的关键字段发生更新后子表如何更新？当主表的记录被删除，子表中已有的匹配记录是否也被删除？当向子表中插入新记录、而主表中不存在匹配的关键字段值时，是否允许子表插入？这些均是表间参照完整性要解决的问题。

参照完整性就是设置一组规则，当发生记录更新、插入或删除等写操作时，以主表为参照，控制子表如何操作。设置参照完整性有 3 个规则：更新规则、删除规则和插入规则，每个规则都有 2~3 个选项：级联、限制和忽略，具体含义如表 2-17 所示。

表 2-17 数据库表间参照完整性规则说明

分 类	规 则 功 能	选 项	选 项 说 明
更新规则	当改变主表中记录的关键字时，子表的记录将如何处理	级联	用主表新的关键字的值更新子表的相关记录
		限制	若子表中有相关记录，则禁止更新主表中相关字段的值
		忽略	不管子表中是否有相关记录，都允许更新主表中相关字段的值
插入规则	当在子表中插入一条新记录或更新一条已经存在的记录时，主表对子表的动作产生何种回应	限制	若主表中不存在匹配的关键字段值，则禁止在子表中插入
		忽略	无论主表中是否存在匹配的关键字段值，都允许在子表中插入
删除规则	当主表中的记录被删除时，子表中相关记录如何处理	级联	在主表中删除记录时，子表中所有相关记录都被删除
		限制	当主表删除记录时，若子表中存在相关记录，则禁止删除
		忽略	当主表中删除记录时，不管子表中是否存在相关记录，都允许删除主表中的记录



实施步骤

1. 在不同工作区同时打开多个表

在数据库设计器中打开数据库 StudentDB.dbc，选择“窗口”“数据工作期”命令，弹出“数据工作期”对话框如图 2-81 所示。

单击“打开”按钮，弹出“打开”对话框，如图 2-82 所示，选择数据表 Student，单击“确定”按钮关闭当前对话框，并在“数据工作期”对话框的“别名”列表框中显示已经打开的数据表 Student，并在状态栏中显示“工作区：1”，结果如图 2-83 所示。



图 2-81 “数据工作期”对话框



图 2-82 “打开”对话框

依次在工作区 2 中打开表 SelCourse，在工作区 3 中打开表 Course。

2. 建立临时关系

以 Student 为主表，建立与表 SelCourse 之间一对多的临时关系。以 Course 为主表，建立与表 SelCourse 之间一对多的临时关系。

通常使用两个表的公共字段作为关键字建立表间的关系。在建立表间关系之前，主表 Student 将“学号”字段设置为主索引字段，子表 SelCourse 将“学号”字段设置为普通索引字段。

在“数据工作期”对话框中的“别名”列表框中，先选择作为主表的表选项 Student，再单击“关系”按钮，将其显示到“关系”列表框中。

在“别名”列表框中单击作为子表的表选项 SelCourse，弹出“设置索引顺序”对话框，如图 2-84 所示选择建立关系的公共索引字段“学号”，单击“确定”按钮，弹出“表达式生成器”对话框。



图 2-83 处于打开状态的表 Student



图 2-84 “设置索引顺序”对话框

在“表达式生成器”对话框中输入建立表关系的关键字表达式“学号”，单击“确定”按钮，如图 2-85 所示，然后返回“数据工作期”对话框。图 2-86 显示建立好了的两表之间的临时关系。



图 2-85 建立表间关系的关键字表达式



图 2-86 建立好了的表间临时关系

3. 浏览建立了临时关系的两个表的数据

在“数据工作期”对话框中的“别名”列表框中选择表 Student，点击“浏览”按钮，在工作区 1 里打开浏览窗口。在工作区 2 里打开表 SelCourse。

观察浏览结果：当主表 Student 的记录指针变化时，子表 SelCourse 将显示与主表当前记录相同学号的所有记录，如图 2-87 所示。

以相同方法建立主表 Course 与子表 SelCourse 之间以“课程编号”为公共关键字的临时关系。

以上 3 个步骤是使用交互方式创建表之间临时关系，下面将使用命令创建临时关系。

在“命令”对话框中输入如下命令：

```
OPEN DATABASE StudentDB
SELE 1
USE Student ALIAS Stu ORDER TAG 学号
SELE 2
USE SelCourse ALIAS SelC_1 ORDER TAG 学号
SELE 1
SET RELATION TO 学号 INTO SelC_1
SET SKIP TO SelC_1
BROWSE FIELDS stu.学号,姓名,性别,专业,SelC_1.课程编号,SelC_1.总成绩
SELE 3
USE COURSE ALIAS Cou ORDER TAG 课程编号
SELE 4
USE SelCourse ALIAS SelC_2 ORDER TAG 课程编号 AGAIN
SELE 3
SET RELATION TO 课程编号 INTO SelC_2
SET SKIP TO SelC_2
BROWSE FIELDS SelC_2.学号,SelC_2.课程编号,课程名称,SelC_2.总成绩
CLOSE TABLES ALL
```

Student						
学号	姓名	出生日期	性别	专业	籍贯	
120201	金明	02/09/99	女	电子技术	吉林	
120202	唐莎	10/10/98	男	电子技术	吉林	
120203	王敏敏	01/20/99	女	电子技术	黑龙江	
120204	李璐	07/10/98	女	电子技术	山东	

Selcourse						
学生学号	课程编号	授课学期	所得学分	课时	期中成绩	期
120202	k007	2	4	72	56.0	
120202	k008	1	8	144	60.0	
120202	k009	4	4	72	93.0	

图 2-87 子表保持与主表当前记录相同学号的所有记录

CLOSE DATABASE

4. 创建永久关系

创建自然情况表 Student 与选修数据表 SelCourse 之间一对多的永久关系 ;创建课程情况表 Course 与选修数据表 SelCourse 之间一对多的永久关系。

数据表之间的临时关系会随着表的关闭而解除,而永久关系在建立时是作为数据库的一部分被保存起来。在后面单元将要介绍的查询设计器、视图设计器或数据环境设计器中,永久关系均被作为表间的默认链接。

以表 Student 作为主表,查看“学号”是否为主索引字段;以表 SelCourse 为子表,查看“学号”是否为普通索引字段。

在数据库设计器中,将表 Student 的主索引“学号”拖动到表 SelCourse 的普通索引“学号”上,两表之间出现一条表示关系的连线,在表 Student 的一端是十字头图标(代表“1”),在表 SelCourse 的一端是三叉头图标(代表“M”),该永久关系是一对多关系。

将表 Course 的主索引“课程编号”拖动到表 SelCourse 的普通索引“课程编号”上,两表之间也出现了一条表示关系的连线,该永久关系是建立在公共字段“课程编号”上的一对多关系。

5. 创建表间参照完整性

创建数据表之间的参照完整性前必须先建立表间的永久关系。

选择“数据库” “清理数据库”命令。通过清理数据库,删除数据库中所有带删除标记的记录。

选择“数据库” “编辑参照完整性”命令,弹出“参照完整性生成器”对话框,编辑更新规则、删除规则和插入规则,如图 2-88 所示。

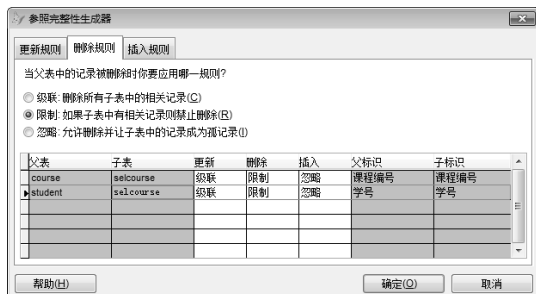


图 2-88 “参照完整性生成器”对话框



阅读材料

1. 与文件相关的测试函数

(1) 表结束测试函数

函数格式:

EOF([数值表达式|表别名])

功能:测试指定工作区或当前工作区中数据表文件记录指针是否指向文件结束(即文件尾)位置。若指向文件尾,函数返回逻辑真(.T.),否则返回逻辑假(.F.)。

参数:

[数值表达式|表别名]:可选项。指定测试工作区号或表别名;若省略,则测试当前工作区数据表。

(2) 表起始测试函数

函数格式:

BOF([数值表达式|表别名])



功能:测试指定工作区或当前工作区中数据表文件记录指针是否指向文件起始(即文件首)位置。若指向文件首,函数返回逻辑真(.T.),否则返回逻辑假(.F.)。

参数:同 EOF()函数。

(3) 当前记录号测试函数

函数格式:

RecNO([数值表达式|表别名])

功能:测试指定工作区或当前工作区中数据表中的当前记录号。假设测试的数据表共有 N 条记录,函数的返回值情况如表 2-18 所示。

参数:同 EOF()函数。

表 2-18 RecNO()函数的返回值情况

序 号	返 回 值	说 明
1	$0 < n < N$	返回当前表或指定工作区中表的当前记录号
2	1	当前记录指针指向文件首时
3	$N+1$	当前记录指针指向文件尾时
4	0	没有打开的数据表时

(4) 记录数测试函数

函数格式:

RecCount([数值表达式|表别名])

功能:测试指定工作区或当前工作区中数据表的记录个数,包括已被逻辑删除的记录。若没有打开的数据表,返回值为 0。

参数:同 EOF()函数。

(5) 文件测试函数

函数格式:

File(<字符表达式>)

功能:测试指定的磁盘文件是否存在,若存在,返回值为真(.T.),否则返回值为假(.F.)。

参数:

<字符表达式>:必选项。指定文件所在的磁盘路径及文件名。

(6) 检索结果测试函数

函数格式:

Found()

功能:测试定位命令的定位是否成功,若成功,返回值为真(.T.),否则返回值为假(.F.)。这些定位命令包括 LOCATE、CONTINUE、SEEK 和 FIND。

(7) 记录删除测试函数

函数格式:

Delete([数值表达式|表别名])

功能:测试指定工作区或当前工作区中数据表的当前记录是否有删除标记,即是否被逻辑删除。若有删除标记,返回值为真(.T.),否则返回值为假(.F.)。

参数:同 EOF()函数。

(8) 表文件名测试函数

函数格式:

DBF([数值表达式|表别名])

功能：测试指定工作区或当前工作区中数据表所在磁盘的完整路径及表名。

参数：同 EOF() 函数。

(9) 字段数测试函数

函数格式：

FCOUNT([数值表达式|表别名])

功能：测试指定工作区或当前工作区中数据表的字段数。

参数：同 EOF() 函数。

2. 数据完整性

数据完整性操作即为了保证数据库中数据的正确性而进行的操作。数据完整性包括实体完整性、域完整性和参照完整性 3 方面，数据库软件系统提供了这 3 类操作规则。

(1) 实体完整性

实体完整性是指在关系中主关键字的值必须唯一且非空，即在数据表中每条记录的主关键字必须有值(非空值)且互不相同，每条记录表示了一个具体的、唯一的实体。在 Visual FoxPro 中可以通过设置主索引字段和候选索引字段来保证不出现重复记录，设置字段的 NULL 属性保证字段值不能为空。例如，“学号”字段、“身份证号”字段均可以设置成主索引或候选索引字段来保证记录的唯一性和实体的完整性。

(2) 域完整性

域完整性是指数据表中字段取值的正确性和有效性。在创建数据表时，用户定义的字段类型、宽度及在创建数据库表时可以定义的“显示”属性和“字段有效性”规则可以保证域的完整性。例如，在输入性别时只能输入“男”或“女”的字符，在输入成绩时，只能输入 0~100 的数值等规则，均属于域完整性规则。

(3) 参照完整性

参照完整性指数据库表之间在建立关系后，通过建立一组规则来约定两个相关联的表之间关联记录的操作规范。Visual FoxPro 提供了 3 类参照完整性规则：插入、更新和删除。这些参照完整性规则控制了在对一个表进行插入、更新和删除操作时，另一个表中相关联记录是否同步进行。例如，在删除规则中选择了“级联”选项，则在任何时候删除父表中的记录时，Visual FoxPro 都会自动删除子表中所有的相关联记录。



项目小结

本项目着重介绍了数据库、数据表的创建、修改、更新、删除、排序、检索的方法及多表之间关系的建立，数据浏览，以及参照完整性的建立。



上机实习

1. 在图书管理数据库 BookDB.dbc 中，按照图 2-89 和图 2-90，以追加的方式为图书表 Book.dbf、读者表 Loander.dbf 输入数据。

图书编号	书名	作者	ISBN号	出版社	价格	存放位置	数量
1	C语言程序设计	李静	9785040354676	电子工业出版社	12.0000	书架	10
2	计算机原理	张雪杨	9781233234323	机械工业出版社	0.0000	书架	20
3	机械基础	黎晨	9786543564758	清华大学出版社	25.0000	书架	10
4	自动控制	董绪方	978656434355	北京工业大学出版社	35.0000	书架	10
5	逻辑电路	秦晓杨	9786765676666	电子工业出版社	13.0000	书架	10
6	电工电子	沙丽	9786544433356	科学出版社	20.0000	仓库	10
9	局域网组建	何东旭	9783433432345	电子工业出版社	34.0000	书架	20

图 2-89 图书表 Book.dbf

读者编号	姓名	证件号码	出生年月	教育	电话	单位	职务	照片
1	王生恩	114252632221456548	12/01/74	研究生	13308789934	电子科技学院	科员	nemo
2	李想	546744324434523345	05/12/76	本科	13654556765	住房资金管理中心	科员	nemo
3	刘璇	343443323434567445	12/04/89	本科	13876768998	城市建设管理中心	科员	nemo
4	齐辉域	544433223422356362	07/09/86	研究生	13309785432	铁路客服中心	助理	nemo
5	于倩	632132455465645677	03/04/74	研究生	15898721231	省中心医院	主任	nemo

图 2-90 读者表 Loander.dbf



2. 以命令方式进行如下操作。

打开图书管理数据库 BookDB.dbc。

打开读者借书表 Loand.dbf。

在读者借书表 Loand.dbf 中追加数据, 如图 2-91 所示。

将该表结构复制到新表 Loand_bak.dbf 中。

将该表数据中读者编号<3 的读者借书记录添加到表 Loand_bak.dbf 中。

关闭所有表。

关闭数据库 BookDB.dbc。

借书序号	读者编号	图书编号	借书时间	还书时间	是否续借	滞纳金
1	1	1	12/03/13	10/02/14	F	\$2.0000
2	2	3	04/01/14	05/31/14	F	\$4.0000
3	4	2	11/17/12	08/03/14	T	\$0.0000
4	5	6	09/01/10	12/01/13	F	\$10.0000
5	3	5	10/07/13	10/01/14	F	\$0.0000
6	2	9	11/08/13	08/09/14	T	\$0.0000
7	1	4	01/12/12	10/01/14	F	\$10.0000

图 2-91 读者借书表 Loand.dbf

3. 参照表 2-13 所示表结构, 在输入读者借书表 Loand.dbf 数据时观察以下各项内容, 并分析原因。

字段“借书编号”、“续借”在浏览窗口中的标题是什么? 为什么?

“借书编号”、“读者编号”和“图书编号”是否可以负数?

字段“续借”在新追加的空记录中是否有初始值? 为什么?

字段“滞纳金”的数据格式是什么? 如何设置?

当输入的“借书时间”>“还书时间”时, 系统会提示什么信息? 如何设置?

4. 打开图书管理数据库 BookDB.dbc 中的图书表 Book.dbf, 在浏览窗口中完成如下任务。

将字段的显示顺序调整为图书编号、书名、作者、ISBN 号、价格、数量、存放位置。

显示记录号价格大于 15 元并且数量小于等于 10 册的记录。

显示字段为“图书编号”、“书名”、“价格”、“作者”的所有记录。

使用命令浏览同时满足条件 (2) \ (3) 的记录。

5. 打开图书管理数据库 BookDB.dbc, 使用菜单功能对表数据完成如下修改。

将读者表 Loander.dbf 中读者编号大于 0 且小于 3 的记录中“单位”字段值更新为“新洲机械厂”, 读者编号大于 3 且小于 8 的记录中“单位”字段值更新为“海滨市税务局”。

将图书表 Book.dbf 中所有记录的“数量”字段值都增加 100。

6. 使用命令完成上题中的修改内容。

7. 使用命令和菜单两种方法分别完成下述内容。

打开图书管理数据库 BookDB.dbc 中的图书表 Book.dbf。

将光标定位在第 2 条记录上。

逻辑删除当前记录。

物理删除所有被逻辑删除的记录。

逻辑删除从当前记录开始向后 5 条记录 (包含当前记录) 中图书数量>10 的所有记录。

恢复所有被逻辑删除的记录。

将当前数据表复制到 Book_bak.dbf 中。

清空当前数据表中的全部记录。

将数据表 Book_bak.DBF 中价格大于 10 元的记录追加到当前数据表中。

关闭所有数据表 and 数据库。

8. 在建立的图书管理数据库 BookDB.dbc 中, 图书表 Book.DBF、读者表 Loander.dbf 和读者借书表 Loand.dbf 都分别建立了索引字段, 分别以交互方式完成按索引字段的排序显示。

将图书表 Book.dbf 中的所有记录按照图书编号升序显示。

将读者表 Loander.dbf 中的所有记录按照读者编号降序显示。

将读者借书表 Loand.dbf 中的所有记录按照借书编号升序显示。

9. 打开图书管理数据库 BookDB.dbc, 使用命令为图书表 Book.dbf 建立索引。

建立基于“价格”字段的单项索引文件 Price.idx。

建立基于“价格”字段的结构复合索引文件, 索引标识为 Prc, 并按降序排列显示。

在表文件的结构复合索引文件中, 按“单位+读者编号”建立索引, 索引标识为 CpnNO, 按升序排列显示。

在表文件中, 按“教育+出生年月”建立一个非结构复合索引文件 EduBirth_Book.cdx, 索引标识为 EduBirth, 按升序排列显示。

10. 打开图书管理数据库 BookDB.dbc, 建立多表之间的关系。

在工作区 1 中打开图书表 Book.dbf, 别名为 Book, 在工作区 2 中打开读者表 Loander.dbf, 别名为 Loander, 在工作区 3 中打开读者借书表 Loand.dbf, 别名为 Loand。

以图书表 Book.dbf 为主表, 读者借书表 Loand 为子表, 建立以“图书编号”为公共索引字段的一对多临时关系, 并浏览两个表的数据。

以读者表 Loander 为主表, 读者借书表 Loand 为子表, 建立以“读者编号”为公共索引字段的一对多临时关系, 并浏览两个表的数据。

创建图书表与读者借书表之间一对多的永久关系; 创建读者表与读者借书表之间一对多的永久关系; 创建如表 2-19 所示的参照完整性。

表 2-19 图书表与读者借书表间的参照完整性设置

父 表	子 表	更新规则	删除规则	插入规则
Book	Loand	级联	级联	限制
Loander	Loand	级联	限制	忽略



知识巩固

一、选择题

1. 在数据输入过程中, 当输入备注型或通用型字段时, 只要在该字段处双击或直接按组合快捷键 (), 即可弹出“数据编辑”对话框。

- A. Ctrl+End B. Ctrl+Delete C. Ctrl+Home D. Ctrl+PageUp

2. 在 Visual FoxPro 中, 以下叙述正确的是 ()。

- A. 自由表的字段可以设置输入掩码
B. 数据库表的字段可以设置输入掩码
C. 自由表的字段都可以设置默认值
D. 自由表和数据库表的字段均不可以设置默认值

3. 在 Visual FoxPro 中, 自由表 ()。

- A. 不可以加入数据库中
B. 可以加入数据库中
C. 加入数据库中后不可以被移出



- D. 只有在打开时才可以加入数据库中
4. 下列说法中正确的是 ()。
- A. 表数据输入后, 表结构不能再改变
 - B. 表数据的输入只能追加到末尾, 不能在记录中插入记录
 - C. 不能同时打开多张表进行操作
 - D. 从另一个表文件中追加记录的命令是 APPEND FROM <表文件名>
5. 如果需要给当前表增加一个字段, 应使用的命令是 ()。
- A. BROWSE
 - B. EDIT
 - C. APPEND
 - D. MODIFY STRUCTURE
6. 当前记录号为 3, 将当前记录设置为第 5 条的命令是 ()。
- A. SKIP 2
 - B. SKIP -2
 - C. SKIP 5
 - D. SKIP -5
7. 利用 LOCATE 命令查找满足条件的第一条记录后, 连续执行 () 命令即可以找到满足条件的其他记录, 直到遇到 () 标志为止。
- A. NEXT, EOF
 - B. CONTINUE, EOF
 - C. GO, BOF
 - D. SKIP, BOF
8. 在 Visual FoxPro 中, APPEND BLANK 命令的功能是 ()。
- A. 在表的第一条记录处添加新记录
 - B. 在当前记录后添加一条新记录
 - C. 在表的末尾添加一条新记录
 - D. 在当前记录前插入一条新记录
9. 假设表文件及其索引文件已打开, 为了使指针定位在物理记录号为 5 的记录上, 应该使用 () 命令。
- A. SKIP 5
 - B. SKIP -5
 - C. GO 5
 - D. GO -5
10. Visual FoxPro 的 ZAP 命令可以完成的操作是 ()。
- A. 物理删除表结构和所有记录
 - B. 逻辑删除所有记录
 - C. 物理删除所有被逻辑删除的记录
 - D. 清空所有记录
11. 顺序执行下列命令后, 最后一条命令的输出结果是 ()。
- ```
USE COURSE
GO 8
SKIP -3
? RecNO()
```
- A. 6
  - B. 5
  - C. 4
  - D. 3
12. 要显示学生自然情况表中出生日期在 1985 年以后 (包含 1985 年) 的学生记录 (出生日期字段的类型为日期型), 使用的命令是 ( )。
- A. DISP FOR 出生日期 >= 1985
  - B. DISP FOR Year (出生日期) >= 85
  - C. LIST FOR Year (出生日期) >= 1985
  - D. LIST FOR Year (出生日期) >= 85
13. 下面关于索引的描述中正确的是 ( )。
- A. 建立索引后, 原来数据表中记录的物理存储顺序被改变
  - B. 索引只能在一个字段中建立



- C. 索引和数据表中的数据存储在一个文件中  
D. 一张表可以建立多个索引, 索引建立后能提高查询效率
14. 建立索引的字段值不允许重复, 并且一张表中只能创建一个的索引类别是 ( )。  
A. 主索引      B. 候选索引      C. 唯一索引      D. 普通索引
15. 可以随着表文件的打开而自动打开的索引文件是 ( )。  
A. 主索引      B. 单向索引      C. 非结构复合索引      D. 结构复合索引
16. 建立主表和子表之间的临时关系时, 若公共关联字段在主表和子表中均是主索引, 则建立的关系为 ( )。  
A. 一对一      B. 一对多      C. 多对一      D. 多对多
17. 设置主控索引的命令是 ( )。  
A. INDEX ON <索引字段>      B. SET INDEX TO <索引文件名>  
C. SET ORDER TO TAG <索引字段>      D. SET RELATION TO <关键字>
18. 表之间建立永久关系后便可以设置参照完整性, 下列叙述中正确的是 ( )。  
A. 表的参照完整性包含更新规则、删除规则、插入规则和新建规则  
B. 在更新规则中, “级联”的含义是将主表中更新的关键字值更新子表中的所有相关记录  
C. 在更新规则中, “限制”的含义是若子表中有相关记录, 允许主表更新关键字值而子表不做修改  
D. 在更新规则中, “忽略”的含义是不管子表中是否有相关记录, 不允许主表更新
19. 下列叙述中错误的是 ( )。  
A. 命令 SELECT 0 执行后选择当前工作区为 0。  
B. 临时关系建立后, 选择主表的一条记录后, 子表就会显示与索引关键字相同的所有匹配记录  
C. 在建立永久关系前, 子表必须以公共字段建立索引  
D. 建立临时关系后, 可以使用 BROWSE 命令浏览多表之间的关系数据
20. 下列叙述中, 错误的是 ( )。  
A. 可以选择字段浏览表数据, 但是不能改变字段的显示顺序  
B. 可以通过条件筛选浏览的记录  
C. 可以在浏览窗口改变字段的显示标题  
D. 在当前工作区中可以使用其他工作区打开的表数据
21. 能够检测记录指针是否在文件尾的函数是 ( )。  
A. Found()      B. BOF()      C. EOF()      D. RecNO()
22. 下列选项中不属于表示范围的关键字是 ( )。  
A. ALL      B. NEXT      C. REST      D. RECNO
23. 索引字段的类型不可以是 ( )。  
A. 数值型      B. 逻辑型      C. 备注型      D. 货币型
24. 逻辑删除记录的命令是 ( )。  
A. DELETE      B. RECALL      C. PACK      D. ZAP
25. 可以批量更新字段值的命令是 ( )。  
A. REINDEX      B. BROWSE      C. EDIT      D. REPLACE



26. 工作区 2 中打开表的别名为 Student, 当前工作区为 1, 在当前工作区中访问表 Student 的字段“学号”的引用方法不正确的是( )。

A. Student.学号    B. A.学号    C. Student- 学号    D. B- 学号

27. 当输入表数据时, 下列说法中错误的是( )。

A. 字段有效性的检测是在光标离开字段时触发  
B. 记录有效性检测是在光标离开有效表达式中包含的相关字段时触发  
C. 记录有效性检测是在光标离开当前记录时触发  
D. 当违反有效性设置时提示的信息可以由用户设置

28. 下面对检索命令 SEEK 和 LOCATE 描述不正确的是( )。

A. 检索结果都可以通过函数 Found()显示  
B. SEEK 是索引查询命令, 必须先建立检索的索引字段  
C. SEEK 和 LOCATE 命令的检索效率一样  
D. LOCATE 只能检索满足条件的第一条记录

29. 能唯一标识一条记录的索引字段可以是( )。

A. 主索引和候选索引    B. 主索引和唯一索引  
C. 唯一索引和普通索引    D. 候选索引和唯一索引

30. 结构复合索引文件的扩展名是( )。

A. .cdx    B. .idx    C. .ndx    D. .fpt

## 二、填空题

1. 数据表首记录的标识为\_\_\_\_\_, 最后一条记录的标识为\_\_\_\_\_, 文件起始标识为\_\_\_\_\_, 文件结束标识为\_\_\_\_\_。

2. 使用表示记录范围的关键字表达下列各项功能。

全部记录: \_\_\_\_\_; 从当前记录开始向后 5 条记录 (包含当前记录): \_\_\_\_\_; 从当前记录开始至最后一条记录: \_\_\_\_\_; 第 N 条记录: \_\_\_\_\_。

3. 浏览数据表的命令是\_\_\_\_\_, 在该命令中选择浏览字段的关键字是\_\_\_\_\_, 筛选浏览记录的关键字是\_\_\_\_\_。

4. 数据表中对记录的删除有\_\_\_\_\_和\_\_\_\_\_, 被\_\_\_\_\_删除的记录可以通过\_\_\_\_\_命令恢复。

5. 索引可以分为\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_; 索引文件可以分为\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_。

6. 建立索引的命令是\_\_\_\_\_; 打开索引文件的命令是\_\_\_\_\_; 设置主控索引的命令是\_\_\_\_\_。

7. 在工作区中以别名打开表的完整命令格式是\_\_\_\_\_。

8. 建立表间的一对一临时关系的命令是\_\_\_\_\_; 建立表间的一对多临时关系的命令是\_\_\_\_\_。

9. 参照完整性规则包括\_\_\_\_\_规则、\_\_\_\_\_规则和\_\_\_\_\_规则, 在建立参照完整性规则之前必须先建立表间的\_\_\_\_\_关系。

10. 命令 SELECT 0 的功能是\_\_\_\_\_。

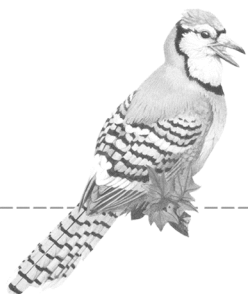
### 三、简答题

1. 简述索引的分类和定义。
2. 表间临时关系和表间永久关系在创建和使用上的区别是什么？
3. 简述参照完整性中不同规则级联、限制和忽略 3 个选项的定义内涵。

### 四、使用命令完成如下任务

在图书管理系统中，打开数据库 BookDB.dbc。

1. 将图书表 Book.DBF 的表结构复制到新表 Book\_bak.dbf 中，并将“高等教育出版社”的记录追加到新表中；浏览新表数据；关闭所有数据表。
2. 打开读者表 Loander.dbf，将读者编号<4 的记录“单位”字段值更新为“沈阳机械厂”。
3. 将记录定位到第 3 条，更新教育程度为“专科”，浏览更新结果。
4. 逻辑删除从第 4 条记录开始向后的 3 条记录。
5. 恢复第 6 条记录，并物理删除其余被逻辑删除的记录。
6. 建立基于“数量”字段的单项索引文件 Count.idx。
7. 建立基于“价格”字段的结构复合索引文件，索引标识为 Price，按降序排列显示。
8. 在表文件的结构复合索引文件中，按“出版社+STR(图书编号,6)”建立索引，索引标识为 PubP，按升序排列显示。
9. 在表文件中，按“出版社+ISBN 号”建立一个非结构复合索引文件 StuSx.cdx，索引标识为 PubISBN，按降序排列显示。
10. 设置当前主控索引为复合结构索引文件中索引标识为 PubP 的索引，并排序显示。
11. 使用当前主控索引查询符合“电子工业出版社”的记录，并使用函数测试查询结果，如果找到，输出记录号。
12. 设置当前主控索引为单项索引文件中索引字段“数量”，并排序显示。
13. 关闭所有索引文件。
14. 检索数量小于等于 10 的图书记录，使用函数显示检索结果，如果找到，输出记录号。



# 单元三

## 数据查询和视图的操作

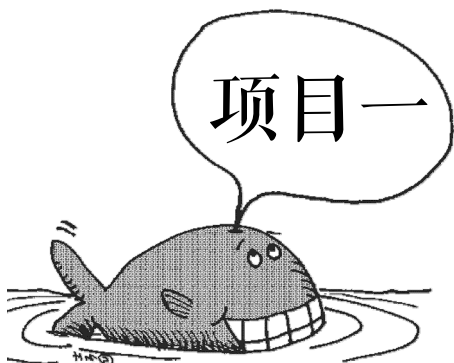
### 知识目标

- 1) 了解查询和视图的基本概念。
- 2) 了解查询和视图的区别。
- 3) 掌握创建查询和视图的方法。
- 4) 熟练掌握 SELECT-SQL 语句中各子句的功能及用法。

### 技能目标

- 1) 能够使用查询设计器建立、运行查询。
- 2) 能用灵活运用 SELECT-SQL 语句建立单表查询、多表查询、条件查询、分组查询、排序查询。
- 3) 能够建立本地视图和远程视图，并更新数据。





# 使用查询设计器

## 项目说明

虽然通过 LOCATE、SEEK 等命令和函数可以实现简单的数据查询，但是要胜任更复杂的查询任务就必须使用 Visual FoxPro 提供的查询功能。使用查询设计器可以根据用户的需求建立查询文件，它既可以实现简单条件的查询，也可以实现复杂条件的查询；既可以创建计算字段，也可以设置查询结果的输出去向。查询文件的扩展名是“.qpr”。

使用查询设计器创建查询的一般步骤如下。

向查询设计器中添加查询所需的表或视图。

如果添加了多个表，需要建立表间的联接。

选择查询输出字段。

设置查询条件。

根据需要设置查询结果的排序顺序。

根据需要设置分组查询。

选择结果的输出类型，如表、报表、标签、浏览等。

保存并运行查询文件，显示查询结果。

本项目分 3 步实施。

任务一：单表查询学生基本情况。

任务二：多表查询学生选课情况。

任务三：创建交叉表，查询学生各门课程的成绩。

## 任务一 单表查询学生基本情况

### 任务目标

能够使用查询设计器创建、运行单表查询，并保存查询文件。



## 任务描述

在 Visual FoxPro 环境下使用查询设计器创建单表查询。本任务要求查询所有女学生的基本情况，如图 3-1 所示。其中查询结果包括字段：学号、姓名、出生日期、性别、专业。查询条件是“性别为‘女’的所有学生”，查询结果是按照出生日期进行升序排序。



| 学号     | 姓名  | 出生日期     | 性别 | 专业    |
|--------|-----|----------|----|-------|
| 120105 | 陈婉雪 | 04/27/98 | 女  | 计算机应用 |
| 120204 | 李璐  | 07/08/98 | 女  | 电子技术  |
| 120202 | 唐莎  | 10/10/98 | 女  | 电子技术  |
| 120203 | 王敏敏 | 01/20/99 | 女  | 电子技术  |
| 120203 | 彭新信 | 05/22/99 | 女  | 计算机应用 |
| 120104 | 李丽  | 08/17/99 | 女  | 计算机应用 |

图 3-1 女学生基本情况

## 相关知识

### 1. 查询的含义与功能

如果要反复查找指定数据内容，可以借助 Visual FoxPro 提供的查询功能。使用查询可以使用户快速得到一组需要的记录，而且每条记录只显示用户指定的字段，Visual FoxPro 将根据用户定义的查询在数据库中检索符合条件的所有记录。因此，在建立查询时，用户可以指定需要显示的字段，并设定过滤条件。在运行查询时，只有那些指定的字段和符合过滤条件的记录才能被显示出来。

利用查询可以从一个或多个表中检索用户所需要的数据（只要这些表之间已经建立了正确的关系），而且还可以对查询结果进行分组和排序。查询结果将产生一个独立的数据文件，但它仅可以作为输出使用，也就是说，用户不能修改查询结果，查询结果也不会影响原来的数据文件。

通过查询，用户可以对记录进行分组、排序和数据统计，查询输出可以作为表单或报表的数据来源，因此查询功能为数据的使用提供了极大的灵活性。在 Visual FoxPro 中，利用查询可以完成以下功能。

- 选择字段：指定在查询结果中的输出字段，可以不必包括表中的所有字段。
- 选择记录：可以指定一个或多个条件，只有符合条件的记录才能在查询结果中显示出来。
- 分组和排序记录：可以对查询结果进行分组，并可对记录按指定的字段升序或降序排列。
- 数据统计：可以建立一个计算字段，利用计算字段保存计算结果。计算字段引用一个或多个表中的一个或多个字段建立数值表达式，统计出表中没有的数据。为了在表单和报表中显示计算字段，用户可以建立一个包含计算字段的查询。
- 使用查询结果作为表单或报表的数据源：为了在表单或报表中显示需要的数据，用户可以建立一个条件查询，将该查询的输出结果作为表单或报表的数据源。当用户每次打开表单或打印报表时，该查询将从基表中检索最新数据。

● 建立新表：用户可以将查询结果存储为一个数据文件，以后用户就可以打开这个数据表进行操作。不过，此数据表是一个自由表，必须使用添加操作才能将它添加到当前数据库中。

总之，利用查询可以只查看指定的字段，规定只显示符合条件的记录。当查询的信息不在一张表上时，还可以从多个表中查询。用户可以将查询的结果保存起来，也可以将查询作为表单或报表的数据来源。



## 实施步骤

启动查询设计器。打开“学生成绩管理信息系统”项目文件，在项目管理器窗口的“数据”选项卡中选择“查询”选项，单击“新建”按钮，弹出如图 3-2 所示的“新建查询”对话框，单击“新建查询”按钮，打开查询设计器窗口和“添加表或视图”对话框，如图 3-3 所示。



图 3-2 “新建查询”对话框



图 3-3 “添加表或视图”对话框

在“添加表或视图”对话框中，将学生自然情况表 Student.dbf 添加到查询设计器窗口中，如图 3-4 所示。

选择输出字段。在“字段”选项卡中的“可用字段”列表框中，选择“student 学号”选项，单击“添加”按钮，将其添加到“选定字段”列表框中，依此操作，再分别将“student 姓名”、“student 出生日期”、“student 性别”、“student 专业”4 个字段添加到“选定字段”列表框中，如图 3-5 所示。

有时为了方便阅读，可以设置输出字段的顺序，把常用的或重要的字段排在前面，在“字段”选项卡中，字段出现的顺序决定了查询输出中信息排列的顺序。为了改变查询输出的列队顺序，可上、下拖动位于字段名左侧的移动框。

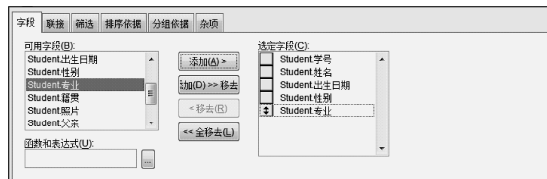


图 3-4 查询设计器窗口



图 3-5 添加字段

确定记录排列顺序。在“排序依据”选项卡中，将“student 出生日期”字段添加到“排序条件”列表框中，“排序选项”默认为“升序”，结果如图 3-6 所示。

“排序条件”列表框中的字段可以是一个，也可以是多个，当有多个字段时，第一个字段决定了主排序的记录顺序，第二个排序字段是在与主子段值相同的记录中再按照该字段进行的排序，依次类推。为了调整排序字段的先后顺序，可以在“排序条件”列表框中，通过字段左侧的移动框将其上下移动，改变排序字段的次序。



设置查询条件。在“筛选”选项卡中设置筛选条件，如图 3-7 所示。



图 3-6 “排序依据”选项卡

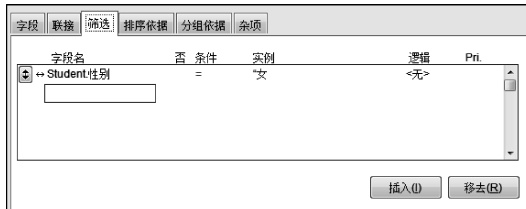


图 3-7 “筛选”选项卡

选项的含义如下。

- 字段名：指定筛选记录的字段，不能选择通用型或备注型字段作为筛选条件。
- 否：单击“否”一栏的按钮，出现“ ”标记，则表示对该行的条件执行取反（NOT）操作。
- 条件：选择关系运算符类型，如=、Like、<、>等。
- 实例：输入相对比的数据实例。在输入时，如果输入的是字符串，需要使用引号定界符，如“男”；如果输入的是日期，需要使用大括号定界符，如{^2014-05-30}；如果是逻辑型数据，则必须使用句点号，如.T.或.F.
- 逻辑：如果包含多个条件查询，使用逻辑运算符 AND 或 OR 将多个条件连接在一起。
- Pri：设置此行筛选条件的优先级别。

设置“杂项”选项卡。如果要使查询结果中没有重复记录，或只返回指定记录数目或百分比数量等，需要在“杂项”选项卡中进行设置，如图 3-8 所示。

设置查询结果的输出方式。当设计完查询后，可以把查询结果输出到浏览窗口或编辑窗口，可以根据不同用途为查询结果指定输出去向。

选择“查询”“查询去向”命令，弹出“查询去向”对话框，如图 3-9 所示。Visual FoxPro 9.0 提供了 4 个查询去向：浏览、临时表、表和屏幕。

- 浏览：在浏览窗口中显示查询结果。
- 临时表：在临时只读表中存储查询结果。
- 表：将查询结果作为表存储。
- 屏幕：在 Visual FoxPro 主窗口或当前输出窗口显示查询结果，如图 3-9 所示。

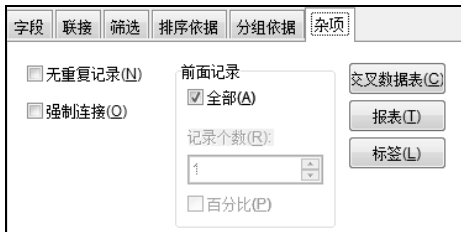


图 3-8 “杂项”选项卡



图 3-9 “查询去向”对话框

运行查询，输出结果如图 3-1 所示。运行查询有如下方式。

- 在项目管理器中，选中“查询 1”，单击“运行”按钮。
- 在查询设计器中，单击工具栏中的“运行”按钮。



- 在查询设计器中，选择“查询” “运行查询”命令。

保存查询文件。单击查询设计器窗口右上角的“关闭”按钮，弹出如图 3-10 所示的对话框。单击“是”按钮，弹出“另存为”对话框，如图 3-11 所示，输入保存文件名（默认文件名为查询 1.qpr）及保存位置，单击“保存”按钮，在项目管理器窗口的“数据”选项卡中，“查询”选项下出现了“查询 1”文件，如图 3-12 所示。



图 3-10 提示对话框

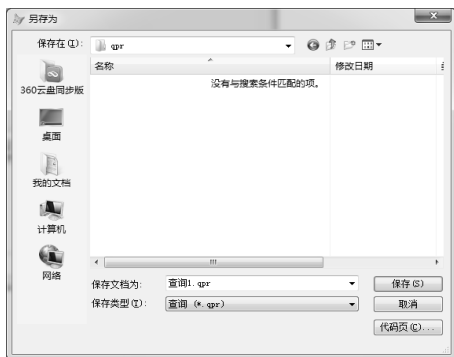


图 3-11 “另存为”对话框

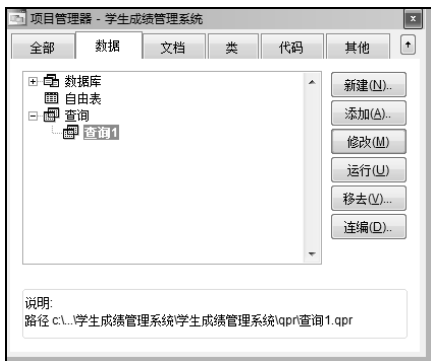


图 3-12 查询文件

## 问题探究

1. 在查询设计器窗口中的“字段”选项卡中，“函数和表达式”列表框的作用是什么？“函数和表达式”列表框的作用是用于创建计算字段。

如果要在查询中显示一个源表中没有的字段，如“平均成绩”，它是由表 SelCourse 中的“期中成绩”字段与“期末成绩”字段的值经过计算形成的，这样的字段称为计算字段，在查询设计器中创建。该计算字段由表达式形成，用来计算数据表中没有的数据，表达式后面接“AS”和“别名”，表示新建一个指定名称的计算字段。

【例 3-1】计算 SelCourse 表中计算机应用专业的学生考试课程的“平均成绩”，结果如图 3-13 所示。

计算公式：平均成绩 = (期中成绩 + 期末成绩) / 2  
操作步骤如下：

新建查询。将表 Student.dbf、Course.dbf 和 SelCourse.dbf 添加到查询设计器中，并确定 3 个表已建立关联。

设置筛选条件。在“筛选”选项卡中设计筛选条件，如图 3-14 所示。

在“字段”选项卡中将表 Student 中的“学号”、“姓名”字段，表 Course 中的“课程编号”字段及表 SelCourse 中的“期中成绩”、“期末成绩”字段添加到“已选定字段”列表框中。在“函数和表达式”文本框中输入表达式 ( SelCourse.期中成绩+SelCourse.期末成绩 )/2 AS 平均成绩，然后将其添加到“已选定字段”列表框中，如图 3-15 所示。

| 学号     | 姓名  | 课程编号 | 期中成绩 | 期末成绩 | 平均成绩    |
|--------|-----|------|------|------|---------|
| 120101 | 杜嘉  | k002 | 0.0  | 0.0  | 0.0000  |
| 120101 | 杜嘉  | k003 | 88.0 | 72.0 | 80.0000 |
| 120101 | 杜嘉  | k004 | 90.0 | 82.0 | 86.0000 |
| 120101 | 杜嘉  | k005 | 0.0  | 0.0  | 0.0000  |
| 120101 | 杜嘉  | k008 | 70.0 | 65.0 | 67.5000 |
| 120102 | 王晓明 | k001 | 30.0 | 0.0  | 15.0000 |
| 120102 | 王晓明 | k002 | 0.0  | 0.0  | 0.0000  |
| 120102 | 王晓明 | k004 | 50.0 | 60.0 | 55.0000 |
| 120102 | 王晓明 | k008 | 66.0 | 77.0 | 71.5000 |
| 120103 | 彭新信 | k002 | 0.0  | 0.0  | 0.0000  |

图 3-13 创建的查询



图 3-14 设置筛选条件

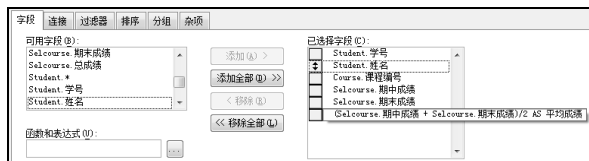


图 3-15 添加计算字段

单击工具栏上的“运行”按钮，浏览查询结果，如图 3-13 所示。其中“平均成绩”是计算字段。

## 2. 如何使用命令创建查询、运行查询和修改查询？

创建查询的命令：

```
CREATE QUERY [FileName|?]
```

【例 3-2】创建一个学生成绩查询文件，存储在 D 盘根目录下。执行以下命令后，系统启动查询设计器。

```
CREATE QUERY D:\XSCJ.QPR
```

运行查询的命令如下：

```
DO <查询文件名>
```

【例 3-3】运行查询文件 XSCJ.QPR，命令如下：

```
DO D:\XSCJ.QPR
```

修改查询的命令如下：

```
MODIFY QUERY <查询文件名>
```

命令执行后，系统自动打开查询设计器。

## 任务二 多表查询学生选课情况

### 任务目标

能够使用查询设计器创建并运行多表查询。

### 任务描述

在 Visual FoxPro 环境下使用查询设计器创建多表查询文件。

本任务要求查询计算机应用专业学生的选课情况，如图 3-16 所示。每个学生显示字段包括学号、姓名、性别、专业、课程编号、课程名称、学分、总成绩。查询结果分别来自学生自然情况表 Student.dbf、学生选修数据表 SelCourse.dbf 及课程情况表 Course.dbf。其中学号、姓名、性别、专业 4 个字段来自表 Student.dbf，课程编号、课程名称、学分 3 个字段值来自表 Course.dbf，总成绩字段来自表 SelCourse.dbf。



### 实施步骤

启动查询设计器。打开“学生成绩管理信息

| 学号     | 姓名  | 性别 | 专业    | 课程编号 | 课程名称  | 学分 | 总成绩  |
|--------|-----|----|-------|------|-------|----|------|
| 120101 | 杜嘉  | 男  | 计算机应用 | k001 | 数据库基础 | 4  | 73.4 |
| 120101 | 杜嘉  | 男  | 计算机应用 | k002 | 计算机基础 | 2  | 0.0  |
| 120101 | 杜嘉  | 男  | 计算机应用 | k003 | 计算机网络 | 4  | 78.4 |
| 120101 | 杜嘉  | 男  | 计算机应用 | k004 | 数据库基础 | 6  | 85.2 |
| 120101 | 杜嘉  | 男  | 计算机应用 | k005 | 专业英语  | 2  | 0.0  |
| 120101 | 杜嘉  | 男  | 计算机应用 | k006 | 高等数学  | 8  | 67.0 |
| 120102 | 王晚明 | 女  | 计算机应用 | k001 | 数据库基础 | 4  | 12.0 |
| 120102 | 王晚明 | 女  | 计算机应用 | k002 | 计算机基础 | 2  | 0.0  |
| 120102 | 王晚明 | 女  | 计算机应用 | k004 | 数据库基础 | 6  | 56.0 |
| 120102 | 王晚明 | 女  | 计算机应用 | k006 | 高等数学  | 8  | 72.6 |
| 120103 | 彭新信 | 男  | 计算机应用 | k002 | 计算机基础 | 2  | 0.0  |
| 120103 | 彭新信 | 男  | 计算机应用 | k004 | 数据库基础 | 6  | 91.2 |

图 3-16 学生选课情况

系统”项目文件，在项目管理器窗口的“数据”选项卡中选择“查询”选项，单击“新建”按钮，弹出如图 3-2 所示的“新建查询”对话框，单击“新建查询”按钮，打开查询设计器窗口和“添加表或视图”对话框，如图 3-3 所示。

在查询设计器窗口中添加表及建立表间联接。将表 Student.dbf、SelCourse.dbf 及 Course.dbf 分别添加到查询设计器中，如图 3-17 所示。由于 3 个表之间没有建立关联，此时会弹出“联接条件”对话框，3 个表之间建立的关联如图 3-18 及图 3-19 所示。



图 3-17 多表关联查询

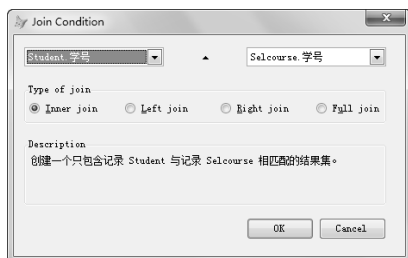


图 3-18 建立表 Student 与表 SelCourse 之间的关联



图 3-19 建立表 Course 与 SelCourse 之间的关联

添加到查询设计器中的多个表如果事先已经建立了关联，则关联表之间会有一条连线。如果表之间没有建立关联，则在添加表时，将会弹出“联接条件”对话框，选择一种联接类型。

两个表之间的联接关系还可以通过查询设计器窗口中的“联接”选项卡进行查看或设置，如图 3-20 所示。

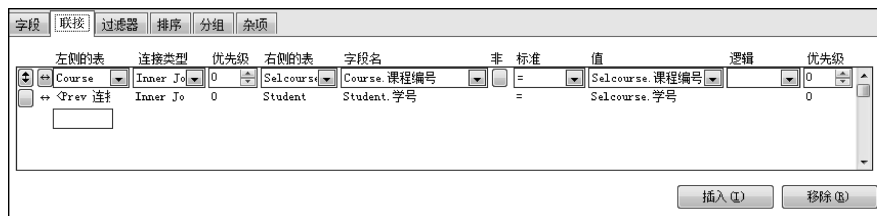


图 3-20 “连接”选项卡

选择输出字段。将“字段”选项卡中的“可用字段”列表框中的“Student. 学号”、“Student. 姓名”、“Student. 性别”、“Student. 专业”、“Course. 课程编号”、“Course. 课程名称”、“Course. 学分”、“SelcCourse. 总成绩”选项全部添加到“已选定字段”列表框中，如图 3-21 所示。



图 3-21 “字段”选项卡



设置查询条件。在“筛选”选项卡中设置筛选条件,如图 3-22 所示。

设置排序顺序。在“排序依据”选项卡中,将“Student.学号”字段添加到“排序条件”列表框中,“排序选项”默认为“升序”,如图 3-23 所示。

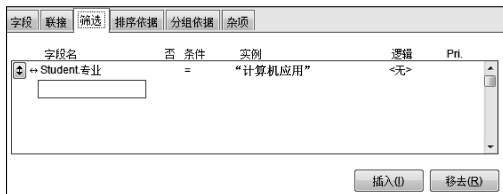


图 3-22 “筛选”选项卡



图 3-23 “排序”选项卡



## 问题探究

### 1. 表间的联接有几种方式?

表间的联接分为内部联接和外部联接,其中外部联接分为左联接、右联接、完全联接。

- 内部联接:在查询结果中,只列出左字段与右字段相匹配的记录,这是默认设置。
- 左联接:在查询结果中,列出左字段列表中的所有记录及右字段列表中与联接条件相匹配的字段内容。
- 右联接:在查询结果中,列出右字段列表中的所有记录及左字段列表中与联接条件相匹配的字段内容。
- 完全联接:在查询结果中,列出两个表中的所有记录,而不考虑记录是否与联接条件相匹配。

例如,有学生表\_a.dbf(如图 3-24 所示)和学生选课表\_c.dbf(如图 3-25 所示),两表的学号有交集,但不能完全一一对应,两张表可以通过“学号”字段相关联,不同的关联方式获得的结果记录集如下。

| 学号     | 姓名  | 出生日期     | 性别 | 专业    |
|--------|-----|----------|----|-------|
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 |
| 120102 | 王晓明 | 11/07/98 | 女  | 计算机应用 |
| 120103 | 彭新信 | 05/22/99 | 男  | 计算机应用 |

图 3-24 源表\_a.dbf 记录集

| 学号     | 课程编号 | 所得学分 | 总成绩  |
|--------|------|------|------|
| 120101 | k001 | 4    | 73.4 |
| 120101 | k002 | 2    | 0.0  |
| 120103 | k002 | 2    | 0.0  |
| 120103 | k004 | 6    | 91.2 |
| 120104 | k003 | 4    | 93.0 |
| 120104 | k005 | 1    | 0.0  |

图 3-25 源表\_c.dbf 记录集

内部联接的 SQL 命令如下:

```
SELECT * FROM a INNER JOIN c ON a.学号=c.学号
```

内部联接后获得的记录结果集如图 3-26 所示,它仅包括两表共有的学号记录。

左联接的 SQL 命令如下:

```
SELECT * FROM a LEFT JOIN c ON a.学号=c.学号
```

左联接后获得的记录结果集如图 3-27 所示,它包含所有表 a 中存在的学号记录。

| 学号_a   | 姓名  | 出生日期     | 性别 | 专业    | 学号_b   | 课程编号 | 所得学分 | 总成绩  |
|--------|-----|----------|----|-------|--------|------|------|------|
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 | 120101 | k001 | 4    | 73.4 |
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 | 120101 | k002 | 2    | 0.0  |
| 120103 | 彭新信 | 05/22/99 | 男  | 计算机应用 | 120103 | k002 | 2    | 0.0  |
| 120103 | 彭新信 | 05/22/99 | 男  | 计算机应用 | 120103 | k004 | 6    | 91.2 |

图 3-26 表 a 与表 c 内部联接后的记录集

| 学号_a   | 姓名  | 出生日期     | 性别 | 专业    | 学号_b   | 课程编号 | 所得学分 | 总成绩  |
|--------|-----|----------|----|-------|--------|------|------|------|
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 | 120101 | k001 | 4    | 73.4 |
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 | 120101 | k002 | 2    | 0.0  |
| 120102 | 王晓明 | 11/07/98 | 女  | 计算机应用 | NULL   | NULL | NULL | NULL |
| 120103 | 彭新信 | 05/22/99 | 男  | 计算机应用 | 120103 | k002 | 2    | 0.0  |
| 120103 | 彭新信 | 05/22/99 | 男  | 计算机应用 | 120103 | k004 | 6    | 91.2 |

图 3-27 表 a 与表 c 左联接后的记录集

右联接的 SQL 命令如下：

```
SELECT * FROM a RIGHT JOIN c ON a.学号=c.学号
```

右联接后获得的记录结果集如图 3-28 所示，它包含所有表 b 中存在的学号记录。

完全联接的 SQL 命令为：

```
SELECT * FROM a FULL JOIN c ON a.学号=c.学号
```

完全联接后获得的记录结果集如图 3-29 所示，它包含表 a 和表 c 中存在的所有学号记录。

| 学号_a   | 姓名   | 出生日期     | 性别   | 专业    | 学号_b   | 课程编号 | 所得学分 | 总成绩  |
|--------|------|----------|------|-------|--------|------|------|------|
| 120101 | 杜嘉   | 03/03/98 | 男    | 计算机应用 | 120101 | k001 | 4    | 73.4 |
| 120101 | 杜嘉   | 03/03/98 | 男    | 计算机应用 | 120101 | k002 | 2    | 0.0  |
| 120103 | 彭新信  | 05/22/99 | 男    | 计算机应用 | 120103 | k002 | 2    | 0.0  |
| 120103 | 彭新信  | 05/22/99 | 男    | 计算机应用 | 120103 | k004 | 6    | 91.2 |
| NULL   | NULL | NULL     | NULL | NULL  | 120104 | k003 | 4    | 93.0 |
| NULL   | NULL | NULL     | NULL | NULL  | 120104 | k005 | 1    | 0.0  |

图 3-28 表 a 与表 c 右联接后的记录集

| 学号_a   | 姓名   | 出生日期     | 性别   | 专业    | 学号_b   | 课程编号 | 所得学分 | 总成绩  |
|--------|------|----------|------|-------|--------|------|------|------|
| 120101 | 杜嘉   | 03/03/98 | 男    | 计算机应用 | 120101 | k001 | 4    | 73.4 |
| 120101 | 杜嘉   | 03/03/98 | 男    | 计算机应用 | 120101 | k002 | 2    | 0.0  |
| 120102 | 王婉明  | 11/07/98 | 女    | 计算机应用 | NULL   | NULL | NULL | NULL |
| 120103 | 彭新信  | 05/22/99 | 男    | 计算机应用 | 120103 | k002 | 2    | 0.0  |
| 120103 | 彭新信  | 05/22/99 | 男    | 计算机应用 | 120103 | k004 | 6    | 91.2 |
| NULL   | NULL | NULL     | NULL | NULL  | 120104 | k003 | 4    | 93.0 |
| NULL   | NULL | NULL     | NULL | NULL  | 120104 | k005 | 1    | 0.0  |

图 3-29 表 a 与表 c 完全联接后的记录集

## 任务三 创建交叉表，查询学生各门课程的成绩

### 任务目标

能够使用查询设计器创建并运行交叉表查询。

### 任务描述

在 Visual FoxPro 环境下使用交叉表向导创建查询文件。

创建一个交叉表，如图 3-30 所示。数据来源于表 SelCourse.dbf，该查询行值显示“学号”字段内容，列值显示“课程编号”字段内容，表中的数据是对应的“总成绩”字段值。

| 学号     | K001 | K002 | K003 | K004 | K005 | K006 | K007 | K008 | K009 | K010 |
|--------|------|------|------|------|------|------|------|------|------|------|
| 120101 | 73.4 | 70.0 | 78.4 | 85.2 | 80.0 |      |      | 67.0 |      |      |
| 120102 | 12.0 | 36.0 |      | 56.0 |      |      |      | 72.6 |      |      |
| 120103 |      | 90.0 |      | 91.2 |      |      |      |      |      |      |
| 120104 | 67.8 |      | 93.0 |      | 80.0 |      |      |      |      |      |
| 120105 |      |      | 95.5 | 77.4 | 60.0 |      |      | 33.0 |      |      |
| 120201 |      |      |      |      |      | 83.8 | 76.2 | 81.2 | 40.0 |      |
| 120202 |      |      |      |      |      |      | 49.4 | 60.0 | 82.8 |      |
| 120203 |      |      | 94.0 |      |      | 90.6 |      | 38.0 |      | 70.0 |
| 120204 |      |      |      |      |      | 69.0 | 87.2 |      | 56.2 | 50.0 |
| 120205 |      |      |      |      |      |      |      | 78.0 | 64.4 | 90.0 |

图 3-30 交叉表查询结果

### 相关知识

所谓交叉表查询就是用行、列的形式汇总数据的查询。使用向导建立交叉表查询时，只能从一个表或视图选择查询输出字段，是基于单表进行的操作。一般是把表中的一个字段值按行输出，另一个字段值按列输出，它们的交叉处输出第三个字段的计算值（如总和、平均数、计数、最大值及最小值等），同时也可以对整行数据进行分类汇总。

创建交叉表查询一般步骤如下：字段选取；定义布局；添加总结信息；完成。



### 实施步骤

打开“学生成绩管理信息系统”项目文件，在项目管理器窗口的“数据”选项卡中选择“查询”选项，单击“新建”按钮，弹出如图 3-2 所示的“新建查询”对话框，单击“查询向导”按钮，弹出“向导选取”对话框，如图 3-31 所示。在“选择要使用的向导”列表框中选择“交叉表向导”选项，单击“确定”按钮，弹出“交叉表向导”对话框。

选择 SelCourse.dbf 表中的“学号”、“课程编号”和“总成绩”字段，如图 3-32 所示。

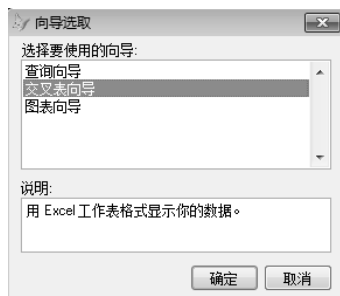


图 3-31 “向导选取”对话框

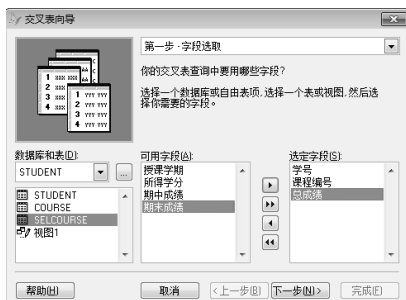


图 3-32 字段选取

单击“下一步”按钮，在弹出的对话框中把“学号”字段拖动到 Row（行）区，把“课程编号”字段拖动到 Column（列）区，把“总成绩”字段拖动到 Data（数据）区，如图 3-33 所示。

单击“下一步”按钮。在这一步，选择交叉表中交叉处理的数据进行何种计算，并把计算结果放到交叉表中，如图 3-34 所示。



图 3-33 定义布局



图 3-34 添加总结信息

- 总结：当交叉处的数据多于一项时，选择计算方式。
- 分类汇总：每一行数据的计算方式。

单击“下一步”按钮完成设置，如图 3-35 所示。

单击“预览”按钮，可预览交叉表查询结果，如图 3-30 所示。单击“完成”按钮，以“交叉表查询.qpr”保存该查询。



## 问题探究

如何利用不同数据源的数据创建交叉表查询？

在使用向导创建交叉表查询时，字段必须来自一个表或视图，如果这些字段不在一个表或视图中，可以先建立一个视图，再利用该视图作为数据源来创建交叉表查询。



图 3-35 完成



## 项目小结

本项目详细介绍了如何根据用户需求运用查询设计器建立查询,既可以是建立简单条件的查询,也可以是建立复杂条件的查询;既可以创建计算字段,也可以设置查询结果的输出去向。通过本项目的学习,读者应该能够创建出需要使用的查询。



## 上机实习

1. 在学生选修课程表 SelCourse.dbf 中,查询全体学生的学号及成绩。
2. 在图书表 Book.dbf 中,查询图书的编号、书名及出版社。
3. 查询读者的信息及所借图书的书名、ISBN 号及借书和还书时间。
4. 查询图书的编号及被借读者的编号。
5. 查询所有读者所借图书的还书时间。



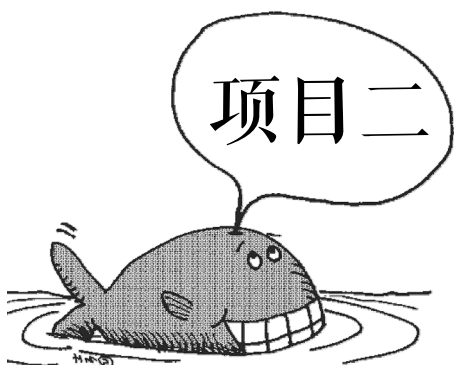
## 知识巩固

### 一、选择题

1. 在下面 4 个文件中,( ) 是查询文件。  
A. cx.prg                  B. cx.bak                  C. cx.qpr                  D. cx.pix
2. 关于查询的正确叙述是 ( )。  
A. 查询与数据库表相同,用来存储数据  
B. 查询结果不能作为表单或报表的数据源  
C. 查询中的数据是可以更新的  
D. 查询是从一个或多个数据库表中导出来为用户定制的虚拟表
3. 在查询设计器中不包括的选项卡是 ( )。  
A. 字段                  B. 筛选                  C. 更新条件                  D. 排序依据
4. 创建查询不能使用的向导是 ( )。  
A. 查询向导                  B. 交叉表向导                  C. 图形向导                  D. 表向导
5. 默认的查询输出去向是 ( )。  
A. 数据表                  B. 图形                  C. 报表                  D. 浏览

### 二、填空题

1. 在项目管理器中运行查询时,先选择\_\_\_\_\_文件,再单击\_\_\_\_\_按钮。
2. 在项目管理器中先选择“查询”选项,再单击\_\_\_\_\_按钮,可以修改选定的查询。
3. 在查询设计器中,“字段”选项卡的作用是\_\_\_\_\_,“联接”选项卡的作用是\_\_\_\_\_,“筛选”选项卡的作用是\_\_\_\_\_,“排序依据”选项卡的作用是\_\_\_\_\_,“分组依据”选项卡的作用是\_\_\_\_\_。
4. 交叉表查询基于\_\_\_\_\_查询,在创建时数据必须来源于\_\_\_\_\_或\_\_\_\_\_。



## 使用 SQL 命令查询

### 项目说明

除了使用查询设计器建立查询以外，Visual FoxPro 还提供了一种 SELECT-SQL 查询语句，为用户提供各种极其简便、快捷和灵活多样的查询功能。

本项目分 5 步实施：

任务一：单表条件查询“选修课程情况”。

任务二：多表关联查询“成绩不及格的学生信息”。

任务三：条件查询“计算机网络课程成绩优秀的女学生”。

任务四：分组查询，统计选课 3 门以上学生的平均成绩和选课门数。

任务五：查询“计算机应用专业全体男生情况”，按出生日期降序排序。

### 任务一 单表条件查询“选修课程情况”

#### 任务目标

使用 SELECT 语句实现单表条件查询。

#### 任务描述

在 Visual FoxPro 环境下使用 SELECT-SQL 查询语句进行单表查询，即从一张表中查询满足条件的记录。

查询选修课程的情况。要求显示表 Course.dbf 中的课程名称、课程编号、考试 3 个字段值，查询的条件是该课程是否为选修课，如果是，“考试”字段值为“.F.”。本任务执行结果如图 3-36 所示。

| 课程编号 | 课程名称  | 考试 |
|------|-------|----|
| k002 | 计算机基础 | F  |
| k005 | 专业英语  | F  |
| k010 | 微机原理  | F  |

图 3-36 “选修课程情况”查询结果



## 相关知识

SELECT-SQL 查询语句是数据库操作中最基本和重要的语句之一,其功能是从数据库中检索满足条件的数据。查询的数据源可以来自一张表,也可以来自多张表甚至视图,查询结果是由多行记录组成的一个记录集合,并允许一个或多个字段作为输出字段。SELECT 语句还可以对查询的结果进行排序、汇总等。其基本使用格式如下:

```
SELECT [ALL | DISTINCT][别名 .] 字段 1 [AS 列标题名] [, [别名 .] 字段 2 [AS 列标题名]]
FROM [数据库名 !] 表名 [[AS] 表别名]
 [INNER | LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER] JOIN [数据库名 !] 表名 [[AS]
表别名] [ON 连接条件] [INNER]
 [WHERE 过滤条件 AND | OR 过滤条件 [AND | OR 过滤条件]]
 [GROUP BY 分组列名 [, 分组列名]]
 [HAVING 过滤条件]
 [ORDER BY 排序依据列]
 [INTO 目标 | TO FILE 文件名 [ADDITIVE] | TO PRINTER | TO SCREEN]
```

1) SELECT 子句: 该子句指出查询结果中显示的字段、常量和表达式。

- “ALL” 为默认设置, 表示查询结果中显示所有行。
- “DISTINCT” 表示从查询结果中除去重复的行, 每个 SELECT 子句中只能使用一次 DISTINCT。

● “别名” 指出表的别名。

● “AS 列标题名” 指出查询结果中显示的列标题。

2) FROM 子句: 该子句指出查询检索数据的表。

● “数据库名!” 指出表所属数据库的名称。

● “[AS] 表别名” 指出表的临时名, 即别名。

● “INNER JOIN” 指出是内连接, 即在查询结果中仅包含与另外的表相匹配的行。

● “LEFT [OUTER] JOIN” 指出查询结果中包含左边表的所有行和右边表中仅匹配的行。

“OUTER” 关键字是任选的, 包含它则强调建立一个外连接。

● “RIGHT [OUTER] JOIN” 指出查询结果中包含右边表的所有行和左边表中仅匹配的行。

● “FULL [OUTER] JOIN” 指出查询结果中包含左边表的所有行和右边表的所有行。

● “ON 连接条件” 指出 FROM 子句中的表的连接条件。

3) WHERE 子句: 指出表之间的关联条件或查询结果中的记录要满足的筛选条件。

● “关联条件” 指出多个查询检索表之间的关联条件。

● “筛选条件” 指出查询结果中的记录必须满足的条件。

4) GROUP BY 子句: 指出基于一个或多个列对查询结果进行分组。

5) HAVING 子句: 指出查询结果中包含的组要满足的条件, 在“过滤条件”中不能包含子查询。

6) ORDER BY 子句: 表示按照一列或多列的数据将查询结果进行排序。

● “ASC” 指出数据按升序排序, 此为默认设置。

● “DESC” 指出数据按降序排序。

7) INTO 子句: 指出查询结果的去向。

INTO DBF 表文件名: 将查询结果保存到一个表文件中。



INTO CURSOR 表文件名：将查询结果保存到一个只读的临时表中。

INTO ARRAY 数组名：将查询结果保存到一个二维数组中。

- 若选择“TO FILE 文件名”，指定查询结果输出到文本文件中，文件名可以带路径。
- 若选择[ADDITIVE]，查询结果追加到文件尾，否则将覆盖原来的内容。
- 若选择“TO PRINTER”，查询结果将会从打印机输出。
- 若选择“TO SCREEN”，指定查询结果输出到屏幕上。

在这些子句中，SELECT 和 FROM 子句是必需的，其他子句都是根据具体的查询需求可选的。

## 任务分析

简单查询是基于单个表，或者是有简单的查询条件的查询。这样的查询由 SELECT 和 FROM 子句构成无条件查询，或者由 SELECT、FROM 和 WHERE 子句构成条件查询。



## 实施步骤

打开“学生成绩管理信息系统”项目文件。

在“命令”对话框中输入如下查询命令。

```
SELECT 课程编号, 课程名称, 考试 FROM Course WHERE Course.考试=.F.
```

执行命令，得到查询结果，如图 3-36 所示。



## 问题探究

### 1. 查询表中全部字段的方法有哪些？

如果要查询表中的全部字段，可以使用两种方法：一种是在 SELECT 子句后列出所有的字段名，另一种是如果列的显示顺序与其在表中定义的顺序相同，则可以简单地在 SELECT 子句后写星号“\*”。例如，查询所有课程的详细信息，命令如下：

```
SELECT * FROM Course
```

### 2. 如何取消取值相同的行？

本来在数据库表中并不存在取值全部相同的记录，但在进行了对字段的选择后，就有可能在查询结果中出现取值完全相同的行。取值相同的行在结果中是没有意义的，因此应删除多余行。

SQL 语言中的 DISTINCT 关键字可以去掉查询结果中的重复行。DISTINCT 关键字放在 SELECT 子句的后边，目标字段名的前边。例如，查询有哪些学生选修了课程，列出选课学生的学号，命令如下：

```
SELECT 学号 FROM SelCourse
```

查询结果如图 3-37 所示。

去掉上述查询结果中重复行的语句如下：

```
SELECT DISTINCT 学号 FROM SelCourse
```

查询结果如图 3-38 所示。

### 3. 如何将查询结果以文本文件的形式保存到磁盘中？

SQL 语句的查询输出结果主要是通过 INTO 子句或 TO 子句来定义，以下将本任务中的单表条件查询“选修课程情况”的结果以不同形式输出。

| 学号     |
|--------|
| 120101 |
| 120101 |
| 120101 |
| 120101 |
| 120101 |
| 120101 |
| 120102 |
| 120102 |
| 120102 |
| 120102 |
| 120103 |
| 120103 |
| 120104 |
| 120104 |

图 3-37 查询结果一

| 学号     |
|--------|
| 120101 |
| 120102 |
| 120103 |
| 120104 |
| 120105 |
| 120201 |
| 120202 |
| 120203 |
| 120204 |
| 120205 |

图 3-38 查询结果二

**(1) 将结果输出到磁盘以表文件形式保存**

```
SELECT 课程编号, 课程名称, 考试 FROM Course ;
WHERE Course.考试=.F. ;
INTO DBF D:\S.dbf
```

**(2) 将结果输出到磁盘以文本文件形式保存**

```
SELECT 课程编号, 课程名称, 考试 FROM Course ;
WHERE Course.考试=.F. ;
TO FILE D:\S.txt
```

**(3) 将结果输出到只读的临时表**

```
SELECT 课程编号, 课程名称, 考试 FROM Course ;
WHERE Course.考试=.F. ;
INTO CURSOR TempDBF
```

**(4) 将结果输出到打印机**

```
SELECT 课程编号, 课程名称, 考试 FROM Course ;
WHERE Course.考试=.F. ;
TO PRINTER
```

## 任务二 多表关联查询“成绩不及格的学生信息”

### 任务目标

使用 SELECT 语句实现多表之间关联查询。

### 任务描述

在 Visual FoxPro 环境下使用 SELECT-SQL 查询语句进行多表条件查询，即从多张表中查询满足条件的记录。

查询成绩不及格的学生信息，结果如图 3-39 所示。

### 相关知识

在数据查询过程中，经常要涉及两个或多个表中的数据，这就需要使用表的连接来实现若干个表数据的联合查询，实现多表查询的方式有多种，可以先建立表之间的关联，再进行查询，也可以用 WHERE 子

| 学号     | 姓名  | 专业    | 课程名称  | 考试 | 总成绩  |
|--------|-----|-------|-------|----|------|
| 120102 | 王婉明 | 计算机应用 | 数据库基础 | F  | 56.0 |
| 120103 | 彭新信 | 计算机应用 | 计算机基础 | F  | 0.0  |
| 120104 | 李丽  | 计算机应用 | 专业英语  | F  | 0.0  |
| 120105 | 顾晓雪 | 计算机应用 | 专业英语  | F  | 0.0  |
| 120105 | 顾晓雪 | 计算机应用 | 高等数学  | F  | 33.0 |
| 120201 | 金明  | 电子商务  | 传感器   | F  | 40.0 |
| 120202 | 唐莎  | 电子技术  | 数字电路  | F  | 49.4 |

图 3-39 成绩不合格学生查询结果



句实现。下面分别进行介绍。

### 1. 建立表之间的关联

在 SQL 中,所有的关联都需要用语句来实现。此时 SELECT-SQL 查询语句的简化格式如下:

```
SELECT 字段名 1, 字段名 2,
FROM 表名 1 INNER | LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER] JOIN 表名 2 ON 关联条件
其他子句
```

### 2. 用 WHERE 子句实现关联

用 WHERE 子句实现关联时,需要将所有的表名写在 FROM 子句后,并在 WHERE 子句中写出表之间的关联条件和查询条件。用 WHERE 子句不能指定关联的类型。

此时 SELECT-SQL 查询语句的简化格式如下:

```
SELECT 字段名 1, 字段名 2,
FROM 表 1, 表 2,
WHERE 关联条件 AND 筛选条件
```

## 任务分析

学生的基本信息、课程成绩分别来自不同的数据库表,因此要想从多个表中组合得到需要的信息,就需要通过多表查询实现。查询“成绩不及格的学生信息”需要将表 Student.dbf、Course.dbf 和表 SelCourse.dbf 进行关联,然后再查询得到结果。



## 实施步骤

打开“学生成绩管理信息系统”项目文件。

在“命令”对话框中输入查询命令,通过建立表之间的关联实现查询的命令格式如下:

SELECT Student.学号,Student.姓名,Student.专业,Course.课程名称,Course.考试,SelCourse.总成绩;

```
FROM Student INNER JOIN SelCourse ON Student.学号=SelCourse.学号;
INNER JOIN Course ON SelCourse.课程编号=Course.课程编号;
WHERE SelCourse.总成绩 <60
```

运用方法二实现查询的命令格式如下:

```
SELECT Student.学号,Student.姓名,Student.专业,Course.课程名称,Course.考试,SelCourse.总成绩;
FROM Student, Course, SelCourse;
WHERE Student.学号=SelCourse.学号 AND SelCourse.课程编号=Course.课程编号 AND SelCourse.总成绩 <60
```

执行命令,得到查询结果,如图 3-39 所示。



## 问题探究

1. 在 SELECT 语句中,字段名前的表名是否可以省略?

在进行多表关联后,在关联生成的表中可能存在字段名相同的列。为了明确需要的是哪个表中的字段列,在字段名前需要添加表名进行限制。

在 SELECT 语句中,“表名.字段名”或者“表名->字段名”的格式指定了字段名及其所属

的表。例如，“Student.学号”或者“Student->学号”表示查询 Student 表中的“学号”字段。其实，对于某个表中独有的字段，命令中可以省略相应的表名。数据库表中“学号”是 Student 表及 SelCourse 表中共有的字段，因此字段名前的表名不可以省略，其他字段如“总成绩”、“专业”、“课程名称”、“考试”等都是该表独有的，因此前面的表名都可以省略。

## 2. 如何实现筛选条件查询？

在查询时对记录的筛选取决于筛选条件。筛选条件是一个包含字段的关系表达式，它跟在 WHERE 子句后面，多个筛选条件根据逻辑关系可以使用 AND（并且）、OR（或者）逻辑运算符相连。例如，查询“……成绩不及格的女生”，使用 WHERE 子句表示如下：

```
.....WHERE 性别=“女”AND 总成绩<60
```

# 任务三 条件查询“计算机网络课程成绩优秀的女学生”

## 任务目标

使用 SELECT 语句实现筛选条件查询。

## 任务描述

在 Visual FoxPro 环境下使用 SELECT-SQL 查询语句进行多表条件查询。通过筛选条件查询“计算机网络课程成绩优秀的女学生”。

## 相关知识

对记录的筛选是通过 WHERE 子句实现的。WHERE 子句后不仅可以跟表的关联条件，同时还可以跟筛选条件。筛选条件是一个逻辑表达式，它是由多个关系表达式通过逻辑运算符（AND、OR、NOT）连接而成的。关系表达式中可以使用的关系运算符如表 3-1 所示。

表 3-1 SELECT-SQL 关系运算符

| 运 算 符                 | 含 义  |
|-----------------------|------|
| =、<、!=、#、==、>、>=、<、<= | 比较大小 |
| AND、OR、NOT            | 多重条件 |
| BETWEEN……AND          | 确定范围 |
| IN、NOT IN             | 确定集合 |
| LIKE、NOT LIKE         | 字符匹配 |
| IS NULL、IS NOT NULL   | 为空值  |

## 1. BETWEEN……AND

这个运算符可以用来查找属性值在（或不在）指定范围内的记录，其中 BETWEEN 指定范围的下限，AND 指定范围的上限。

## 2. IN

当字段中的值与 IN 中的某个常量值相等时，表明此记录为符合查询条件的记录。例如，



家庭住址 IN ('辽宁','吉林')表示查找家庭住址在辽宁和吉林的学生记录。

### 3. LIKE

用于查找指定字段中与匹配串相匹配的记录。匹配串是一种特殊的字符串,其特殊之处在于它不仅包含普通字符,还可以包含通配符。通配符用于表示任意的字符或字符串。

匹配串中可以包含字符常量,也可以包含如下 4 种通配符。

- \_ (下画线): 匹配任意一个字符。
- % (百分号): 匹配 0 个或多个字符。
- [ ]: 匹配 [ ] 中的任意一个字符。例如, [abcd] 表示匹配 a、b、c、d 中的任何一个。
- [^]: 不匹配 [ ] 中的任意一个字符。例如, [^abcd] 表示不匹配 a、b、c、d。

### 4. IS NULL

空值 (NULL) 在数据库中有特殊的含义,它表示不确定的值。例如,某些学生选修课程后还没有考试,则这些学生虽然有修课记录,但没有考试成绩,因此考试成绩即为空值。判断某个字段的值是否为 NULL,不能使用普通的比较大小运算符 (=、!=等),而只能使用专门的判断 NULL 值的子句来完成。

判断字段列取值为空的语句格式如下:

```
字段名 IS NULL
```

判断字段列取值不为空的语句格式如下:

```
字段名 IS NOT NULL
```

### 5. AND、OR、NOT

在 WHERE 子句中使用逻辑运算符 AND 和 OR 来组成多重条件查询。AND 表示满足所有的条件表达式结果才为 TURE, OR 表示只要满足其中一个条件表达式结果即为 TURE。NOT 是逻辑上取反。

## 任务分析

条件查询是通过 WHERE 子句来实现的。本任务的筛选条件有 3 个,分别是:课程名称为“计算机网络”,性别为“女”;总成绩要大于等于 90 分。

只有同时满足这 3 个条件的学生记录才能被显示出来,查询结果如图 3-40 所示。

| 学号     | 姓名  | 性别 | 专业    | 课程名称  | 总成绩  |
|--------|-----|----|-------|-------|------|
| 120105 | 顾晓雪 | 女  | 计算机应用 | 计算机网络 | 95.5 |
| 120203 | 王敬敏 | 女  | 电子技术  | 计算机网络 | 94.0 |
| 120104 | 李丽  | 女  | 计算机应用 | 计算机网络 | 93.0 |

图 3-40 多表条件查询结果



## 实施步骤

打开“学生成绩管理信息系统”项目文件。

在“命令”对话框中输入查询命令,命令格式如下:

```
SELECT Student.学号,Student.姓名,Student.专业,Course.课程名称,Course.考试,SelCourse.总成绩;
FROM Student INNER JOIN SelCourse ON Student.学号=SelCourse.学号;
INNER JOIN Course ON SelCourse.课程编号=Course.课程编号;
WHERE Course.课程名称="计算机网络" AND Student.性别="女" AND SelCourse.总成
```

绩>=90

执行命令，得到查询结果，如图 3-40 所示。



## 问题探究

1. 如何使用 WHERE 子句同时限定关联条件和筛选条件实现本任务？

关联条件和筛选条件可以同时跟在 WHERE 子句的后面，本任务使用这种方法的语句格式如下：

```
SELECT Student.学号,Student.姓名,Student.专业,Course.课程名称,Course.考试,SelCourse.总成绩;
FROM Student,Course,SelCourse;
WHERE Student.学号=SelCourse.学号 AND SelCourse.课程编号=Course.课程编号 AND
Course.课程名称=" 计算机网络 " AND Student.性别=" 女 " AND SelCourse.总成绩>=90
```

## 任务四 分组查询、统计选课 3 门以上学生的平均成绩和选课门数

### 任务目标

使用 SELECT 语句实现分组统计查询。

### 任务描述

在 Visual FoxPro 环境下使用 SELECT-SQL 查询语句进行分组统计查询。

分组查询，统计选修了 3 门以上课程的学生平均成绩和选课门数，查询结果如图 3-41 所示。

### 任务分析

该查询的数据源来自学生选课表 SelCourse.dbf，以学号进行分组，查询选修了 3 门以上课程的学生选课门数和各科平均成绩。先利用 GROUP BY 子句对表 SelCourse.dbf 数据按照学号进行分组，然后再利用统计函数（COUNT()、AVG()）分别对每一组记录（学号相同的记录）进行统计，最后筛选出组记录数大于 3 的组，并计算出各科平均成绩，查询结果如图 3-41 所示。

| 学号     | 选课门数 | 平均成绩  |
|--------|------|-------|
| 120101 | 6    | 75.67 |
| 120102 | 4    | 44.15 |
| 120105 | 4    | 66.48 |
| 120201 | 4    | 70.30 |
| 120203 | 4    | 73.15 |
| 120204 | 4    | 65.60 |

图 3-41 分组查询结果

### 相关知识

#### 1. 统计函数

统计函数也称为聚合函数或集合函数、聚集函数，其作用是对一组值进行计算并返回一个单值。SQL 提供的统计函数如下。

- COUNT (\*): 统计表中元组的个数。
- COUNT ([DISTINCT]<列名>): 统计本列中非空值的个数。DISTINCT 表示不包括列的重复值。



- SUM (<列名>): 计算列值总和 (必须是数值型列)。
- AVG (<列名>): 计算列值平均值 (必须是数值型列)。
- MAX (<列名>): 求列值最大值。
- MIN (<列名>): 求列值最小值。

注意: 统计函数不能出现在 WHERE 子句中。

上述函数中除 COUNT (\*) 外, 其他函数在计算过程中均忽略 NULL 值。统计函数的计算范围可以是满足 WHERE 子句条件的记录 (如果是对整个表进行计算), 也可以对满足条件的组进行计算 (如果进行了分组操作)。

## 2. 对查询结果进行分组统计

除了对全表统计以外, 有时也希望计算的精度更为精确一些。例如, 统计每个学生的考试平均成绩, 而不是全体学生的考试平均成绩, 这就需要用到 SQL 语句的分组统计功能。

在 SQL 语句中实现分组功能的子句是 GROUP BY。GROUP BY 可将统计控制在组一级。分组的目的是细化统计函数的作用对象。在一个 SQL 语句中, 可以使用多个列进行分组。需要注意的是, 如果使用了分组子句, 则跟在 SELECT 子句后的查询列表所包含的列一般都是分组依据列 (即跟在 GROUP BY 子句后边的列) 或是统计函数。

使用 GROUP BY 子句时, 如果在 SELECT 的查询列表中包含统计函数, 则是针对每个组计算出一个汇总值, 从而实现对查询结果的分组统计。

分组子句一般在 WHERE 子句的后边, 它的一般形式如下:

GROUP BY <分组依据列>[, ..... , n]

[HAVING <组提取条件>]

注意:

- GROUP BY 子句中的分组依据列必须是表中存在的列名, 不能使用 AS 子句指派的列别名。

- 带有 GROUP BY 子句的 SELECT 语句的查询列表中只能出现分组依据列或统计函数, 因为分组后每个组只能返回一行结果。

HAVING 子句用于对分组后的结果再进行筛选, 它的功能类似 WHERE 子句, 但它作用于组而不是单个记录。在 HAVING 子句中可以使用统计函数, 但在 WHERE 中则不能。HAVING 子句通常与 GROUP BY 子句一起使用。

正确理解 WHERE、GROUP BY、HAVING 子句的作用及执行顺序, 对于编写正确、高效的查询语句有很大的帮助。

- WHERE 子句用来筛选 FROM 子句中指定的数据源所产生的行数据。
- GROUP BY 子句用来对经 WHERE 子句筛选后的结果数据进行分组。
- HAVING 子句用来对分组后的结果数据再进行筛选。

对于可以在分组操作之前应用的搜索条件, 在 WHERE 子句中指定它们更有效, 这样可以减少参与分组的数据行。在 HAVING 子句中指定的搜索条件应该是那些必须在执行分组操作之后应用的搜索条件。



## 实施步骤

打开“学生成绩管理信息系统”项目文件。



在“命令”对话框中输入查询命令，命令格式如下：

```
SELECT 学号, COUNT(*) AS 选课门数, AVG(总成绩) AS 平均成绩;
FROM SelCourse;
GROUP BY 学号;
HAVING COUNT(*) > 3
```



## 问题探究

1. 在进行统计、分组查询时要注意哪些问题？

当查询语句的目标字段列中包含统计函数时，若没有分组子句，则目标字段列中只能包含统计函数，而不能再添加其他字段。若包含分组子句，则在查询的目标字段列中除了可以包含统计函数外，还可以包含分组依据字段列。

2. 如何对统计后的结果进行筛选？

对统计后的结果进行筛选的条件应该写在 HAVING 子句中，而不应该写在 WHERE 子句中。例如，查询平均成绩大于 60 的学生，因为“平均成绩”为统计字段，若将条件写成“WHERE AVG(总成绩) > 60”是错误的，写成“HAVING AVG(总成绩) > 60”才是正确的。

## 任务五 查询“计算机应用专业全体男生情况”，按出生日期降序排序

### 任务目标

使用 SELECT 语句实现查询结果的排序。

### 任务描述

在 Visual FoxPro 环境下使用 SELECT-SQL 查询语句实现查询结果的排序。

单表条件查询计算机应用专业全体男生情况（包括学号、姓名、出生日期、专业及籍贯），按出生日期降序排序，查询结果如图 3-42 所示。

| 学号     | 姓名  | 出生日期     | 专业    | 籍贯  |
|--------|-----|----------|-------|-----|
| 120102 | 王小明 | 11/07/98 | 计算机应用 | 黑龙江 |
| 120101 | 杜嘉  | 03/03/98 | 计算机应用 | 辽宁  |

图 3-42 排序查询结果

### 相关知识

如果要将查询结果按照一定的顺序显示出来，如按考试成绩从高到低排列学生查询记录，可以通过在 SELECT 命令中加入 ORDER BY 子句来控制查询结果的显示顺序。ORDER BY 子句可以按升序（默认或 ASC）、降序（DESC）排列结果行。

如果 ORDER BY 子句选项中有多个排序项，排序时先按第一个排序项排序，如果第一个排序项的值相同，再按第二个排序项排序，依此类推。

ORDER BY 子句必须是 SQL 命令中的最后一个子句（除 INTO 子句外）。

### 任务分析

这是一个单表的排序查询，查询的数据均来自学生自然情况表 Student.dbf，要求显示学生



的基本情况 (包括学号、姓名、出生日期、专业及籍贯), 查询的条件是专业为 “ 计算机应用 ” 且学生性别为 “ 男 ”。



## 实施步骤

打开 “ 学生成绩管理信息系统 ” 项目文件。

在 “ 命令 ” 对话框中输入查询命令, 命令格式如下:

```
SELECT 学号, 姓名, 出生日期, 专业, 籍贯;
FROM Student;
WHERE 性别= " 男 " AND 专业= " 计算机应用 " ;
ORDER BY 出生日期 DESC
```

执行命令, 得到如图 3-42 所示的查询结果。



## 项目小结

SQL 查询语句是数据库操作中最基本和最重要的语句之一, 运用查询语句能够从数据库中检索满足条件的数据。查询的数据源可以来自于一张表, 也可以来自多张表, 查询的结果是由 0 行 (没有满足条件的数据) 或者多行记录组成的一个记录集合, 并允许选择一个或多个字段作为输出字段。通过本项目的学习, 读者可以依据需求运用 SELECT-SQL 查询语句对数据库中的数据进行查询或修改。



## 上机实习

1. 查询图书表中的全部数据。
2. 查询学生选课表中学生的姓名、年龄。
3. 查询年龄在 25 ~ 45 岁之间的读者的教育程度。
4. 查询借阅图书编号为 x x x x 的读者信息。
5. 查询读者编号为 x x x x 的读者所借图书的信息。
6. 统计每本图书的借阅人数。
7. 统计每本图书的借阅次数, 并按升序显示。
8. 查询 “ 滞纳金 ” 超过 8 元的读者, 要求列出读者姓名及图书名。
9. 查询没有借阅过图书的读者姓名及电话。
10. 查询没被借阅过的图书的书名及作者。



## 知识巩固

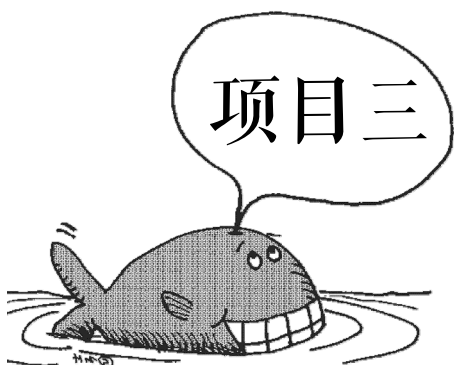
### 一、选择题

1. 在 SQL 语句中, 与表达式 “ 成绩 BETWEEN 80 AND 90 ” 功能相同的表达式是 ( )。  
A. 成绩 >= 80 AND 成绩 <= 90                      B. 成绩 > 80 AND 成绩 < 90  
C. 成绩 <= 80 AND 成绩 > 90                      D. 成绩 >= 80 OR 成绩 <= 90
2. 使用 SQL 语句进行分组检索时, 为了去掉不满足条件的分组, 应当 ( )。  
A. 使用 WHERE 子句  
B. 在 GROUP BY 后面使用 HAVING 子句

- C. 先使用 WHERE 子句,再使用 HAVING 子句  
D. 先使用 HAVING 子句,再使用 WHERE 子句
3. 查询设计器中的“联接”选项卡对应 SQL 语句中的( )子句。  
A. WHERE          B. GROUP BY          C. FROM          D. ORDER BY
4. 在 SQL 的 SELECT 查询结果中,消除重复记录的方法是( )。  
A. 通过指定主关键字          B. 通过指定唯一索引  
C. 使用 DISTINCT 子句          D. 使用 HAVING 子句
5. 在 Visual FoxPro 中,以下关于 SQL 中的 SELECT 语句描述中错误的是( )。  
A. SELECT 子句中可以包含表中的列和表达式  
B. SELECT 子句中可以使用别名  
C. SELECT 子句规定了结果集中的列顺序  
D. SELECT 子句中列的顺序应该与表中列的顺序一致

## 二、填空题

1. 使用 SELECT 语句进行分组计算查询时,可以使用\_\_\_\_\_子句去掉不满足条件的记录。
2. 查询设计器的“排序依据”选项卡对应于 SELECT-SQL 语句的\_\_\_\_\_短语。
3. 在 SELECT-SQL 语句中为了将查询结果存储到永久表应该使用\_\_\_\_\_短语。
4. 在 SQL 语句中,空值用\_\_\_\_\_表示。



## 使用视图更新数据表

### 项目说明

视图是数据库中的一个对象，它是数据库管理系统提供给用户的以多种角度观察数据库中数据的一个重要机制。

视图与查询相似，它们都可以很方便地从一个或多个相关联的表中检索出所需要的数据，但查询是以文件形式单独保存的，而视图是作为数据库的一部分被保存；查询只能检索，不能修改数据，而视图却可以更新表中的数据，并保存更新结果。

视图分为本地视图和远程视图两种。本地视图是在 Visual FoxPro 中使用 SQL 语句从视图和表中提取信息，远程视图是使用远程 SQL 语句从远程 ODBC 数据源表中提取信息。

本项目分三步实施：

任务一：创建本地视图。

任务二：使用视图更新数据。

任务三：创建和运行远程视图。

### 任务一 创建本地视图

#### 任务目标

能够使用视图设计器创建本地视图。

#### 任务描述

以学生自然情况表 Student.dbf 为数据源，使用视图设计器创建“电子技术专业学生情况”本地视图，视图包含的字段列表如图 3-43 所示，并以 ElecStudent

| 学号     | 姓名  | 出生日期     | 性别 | 专业   |
|--------|-----|----------|----|------|
| 120202 | 唐莎  | 10/10/98 | 男  | 电子技术 |
| 120203 | 王敏敏 | 01/20/99 | 女  | 电子技术 |
| 120204 | 李璐  | 07/08/98 | 女  | 电子技术 |
| 120205 | 杨建  | 06/30/99 | 男  | 电子技术 |

图 3-43 创建本地视图

为视图名保存。

## 相关知识

视图和查询有很多相似的功能，包括创建视图所需要的表和字段、筛选符合条件的记录、创建计算字段等。视图是一个虚拟表，视图中的数据是从已有的数据库表或其他视图中选取而得来的。这些数据在数据库中并不实际存储，仅在数据库的数据词典中存储视图的定义。视图一经定义，就成为数据库的一个组成部分，可以向数据库表一样供用户查询数据。

视图不仅可以用于查询信息，而且还可以更新数据。视图具有如下特点。

### (1) 提供附加的安全层

视图是建立在基本表上的一种表，它只允许用户看到视图中所定义的数据而不是视图引用表中的数据，从而提高了数据的安全性。

### (2) 隐蔽数据的复杂性

数据库包含多张数据表，在执行表联接操作时，用户可以从几张表中抽取数据，但这些联接操作最终常常把用户甚至专家也搞糊涂。在这种情况下，就有必要建立混合多表信息的视图。

### (3) 命名简洁

在建立数据库时，一些表的字段名可能很复杂，在建立视图时，可以更改字段名称，以方便用户识别。在视图中换名并不会改变基表的定义。

### (4) 更改灵活性

可更改组成视图的一个或多个表的内容而不用更改引用该视图的应用程序。例如，有一个由两个表联结的视图，从一个表中显示 3 列，而从另一个表中显示 4 列。如果前一个表中增加一列，这对视图的定义不会产生影响，也不会影响涉及此视图的应用。

### (5) 可更新数据库

使用视图不仅可以查询数据库数据，更重要的是可以通过视图更新数据库信息。这一点非常重要，也是视图的突出优点之一。



## 实施步骤

启动视图设计器。打开“学生成绩管理信息系统”项目文件，在 StudentDB 数据库中选择“本地视图”选项，单击“新建”按钮，弹出如图 3-44 所示的“新建本地视图”对话框，单击“新建视图”按钮，打开视图设计器窗口和“添加表或视图”对话框，如图 3-45 所示。



图 3-44 “新建本地视图”对话框

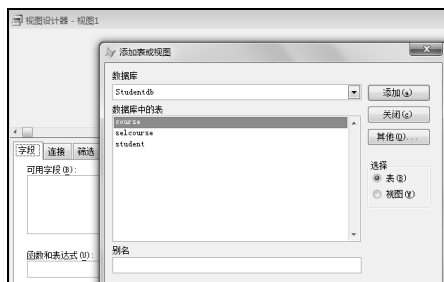


图 3-45 视图设计器窗口及“添加表或视图”对话框

添加表或视图。在“添加表或视图”对话框中，将 Student 表添加到视图设计器中。



选择视图中的字段。在“字段”选项卡中将“student 学号”、“student 姓名”、“student 出生日期”、“student 性别”、“student 专业”字段添加到“选定字段”列表框中，如图 3-46 所示。

设置筛选条件。在“筛选”选项卡中设置筛选条件，如图 3-47 所示。

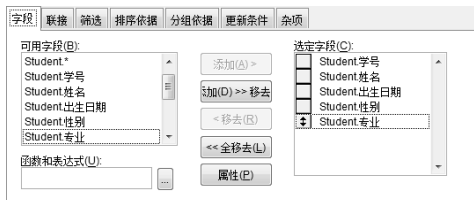


图 3-46 “字段”选项卡

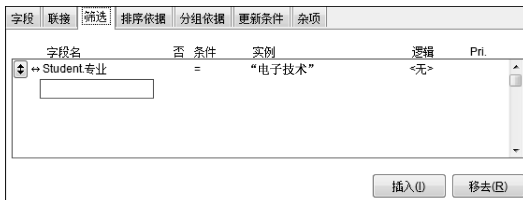


图 3-47 设置筛选条件

单击 按钮，观察运行结果，如图 3-43 所示。

选择“文件”→“保存”命令，弹出如图 3-48 所示的“保存”对话框，以“ElecStudent”保存该视图，保存后的视图成为 StudentDB 数据库的一个对象，在数据库管理器中如图 3-49 所示。



图 3-48 “保存”对话框



图 3-49 包含本地视图的数据库



## 问题探究

如何使用命令创建本地视图？

使用创建视图的 SQL 命令，可以方便地创建本地视图，该命令格式如下：

```
CREATE VIEW <视图名> AS <SELECT 语句>
```

参数：

- “视图名”指创建的视图名称。
- “SELECT 语句”：指定视图显示的数据。

例如，使用命令创建本任务的视图语句如下：

```
CREATE VIEW ElecStudent AS ;
SELECT Student.学号, Student.姓名, Student.出生日期, Student.性别, ;
Student.专业 FROM studentdb!student ;
WHERE Student.专业 = "电子技术"
```

## 任务二 使用视图更新数据

### 任务目标

能够使用本地视图更新表数据。

## 任务描述

使用本地视图 ElecStudent 更改数据源基表 Student.dbf 中“专业”字段的值。


## 相关知识

在视图设计器中，“更新条件”选项卡用于指定更新条件，可以把对数据的修改（如更新、删除和插入）回送到数据源中。



## 实施步骤

打开“学生成绩管理信息系统”项目文件，选中视图 ElecStudent，单击“修改”按钮，打开视图设计器窗口，选择“更新条件”选项卡，如图 3-50 所示。

设置关键字。在“更新条件”选项卡中，单击“字段名”列表框中左侧带有钥匙图标  的一列，出现“ ”标记，将“学号”字段设置为关键字段，如图 3-50 所示。

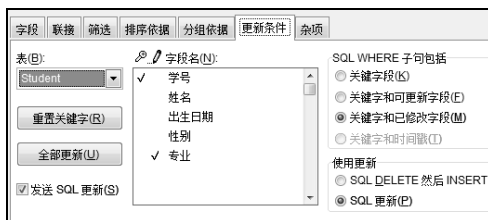



图 3-50 “更新条件”选项卡

为了使表中的字段值能被修改，首先必须为表定义一个关键字段，Visual FoxPro 根据关键字段将在视图中修改的记录与表中的原始记录相匹配。关键字段的设置必须是唯一的，若有重复值，则必须设计多个关键字以避免重复记录，否则系统将无法判断要更新的记录。

设置可更新的字段。在“更新条件”选项卡中，单击“字段名”列表框中左侧带有铅笔图标  的一列，出现“ ”标记，将“专业”字段设置为可更新字段，如图 3-50 所示。

如果字段未被标注为可更新字段，虽然在表单或视图浏览窗口中可以更新修改这些字段值，但修改的值不会被存回到源表中。另外由于关键字段是用于唯一标识每一条记录的，所以最好不要将关键字段设置为可更新的字段。

发送 SQL 更新。如果要把表中更新的记录存回到源表中，必须勾选“发送 SQL 更新”复选框。在使用此选项之前，必须至少设置一个关键字段和一个可更新的字段，如图 3-50 所示。

检查更新冲突。在多用户环境下，远程服务器中的数据可以被多人同时访问。为了检查在视图更新数据之前源表数据是否被他人修改过，也就是比较视图中指定字段的旧值和原始表中该字段的当前值是否相同，如果两个值相同。则认为原始值未做更新，不存在冲突；如果它们不相等，则存在冲突，数据源返回一条错误信息。使用“SQL WHERE 子句包括”选项组可以管理当多个用户访问同一数据时应如何更新记录。

- 关键字段：当源表中的关键字段被修改时，则更新操作失败。
- 关键字和可更新字段：当修改了源表中关键字段或任何可更新的字段时，则更新操作失败。
- 关键字和已修改字段：从视图首次检索以后，如果关键字段或任何被修改过的字段在源表中又被改变，则更新操作失败。
- 关键字和时间戳：只有当远程表有时间戳列时，该选项才有效。“时间戳”是大型数据库的概念，Visual FoxPro 9.0 中没有。



选择更新方式。“使用更新”选项组用来控制系统更新源表数据采用 SQL 命令的方法。

- SQL DELETE 然后 INSERT：先删除记录，然后使用在视图中输入的新值取代源值。
- SQL 更新：使用 SQL UPDATE 命令来更新记录。

利用“更新条件”选项卡对字段进行更新设置后，可在视图浏览窗口修改可更新的字段值。例如，将学号为“120201”学生的专业修改为“电子商务”，如图 3-51 所示。

更改后的 Student 表记录如图 3-52 所示。

| 学号     | 姓名  | 出生日期     | 性别 | 专业    |
|--------|-----|----------|----|-------|
| 120104 | 李丽  | 08/17/99 | 男  | 计算机应用 |
| 120105 | 顾晓雪 | 04/27/98 | 男  | 计算机应用 |
| 120201 | 金明  | 02/09/99 | 女  | 电子技术  |
| 120202 | 唐莎  | 10/10/98 | 男  | 电子技术  |

| 学号     | 姓名  | 出生日期     | 性别 | 专业   |
|--------|-----|----------|----|------|
| 120201 | 金明  | 02/09/99 | 女  | 电子商务 |
| 120202 | 唐莎  | 10/10/98 | 男  | 电子技术 |
| 120203 | 王敏敏 | 01/20/99 | 女  | 电子技术 |
| 120204 | 李璐  | 07/08/98 | 女  | 电子技术 |
| 120205 | 杨建  | 06/30/99 | 男  | 电子技术 |

图 3-51 在视图中修改数据

| 学号     | 姓名  | 出生日期     | 性别 | 专业    | 籍贯  |
|--------|-----|----------|----|-------|-----|
| 120105 | 顾晓雪 | 04/27/98 | 男  | 计算机应用 | 河北  |
| 120201 | 金明  | 02/09/99 | 女  | 电子商务  | 吉林  |
| 120202 | 唐莎  | 10/10/98 | 男  | 电子技术  | 吉林  |
| 120203 | 王敏敏 | 01/20/99 | 女  | 电子技术  | 黑龙江 |
| 120204 | 李璐  | 07/08/98 | 女  | 电子技术  | 山东  |
| 120205 | 杨建  | 06/30/99 | 男  | 电子技术  | 河北  |

图 3-52 更新表后的记录

## 任务三 创建和运行远程视图

### 任务目标

能够使用视图设计器创建远程视图并更新远程数据表中的数据。

### 任务描述

先启动 microsoft Access，创建一个空 Access 数据库文件 MyBookAccess.accdb 或 MyBookAccess.mdb（该数据库可以是本机数据库，也可以是局域网中某台机器上的数据库）。在该数据库中建立一张“图书信息表”，“图书信息表”的结构如表 3-2 所示。

表 3-2 “图书信息表”的结构

| 图书编号（文本型，5） | 图书名称（文本型，20） | 出版社（文本型，20） | 书价（货币型） |
|-------------|--------------|-------------|---------|
| 00001       | 微机计算机原理与应用   | 希望出版社       | ¥34.80  |
| 00002       | 计算机组装与维护     | 高等教育出版社     | ¥24.00  |
| 00003       | 电子商务基础与实务    | 电子工业出版社     | ¥30.50  |

在“学生成绩管理”数据库 StudentDB.dbc 中创建远程视图 RemoteAccess，以 Access 数据表“图书信息表”为数据源，设置更新条件，设置“图书编号”为主键，设置所有字段都可以更新，运行该视图并更新数据源中的数据。

### 相关知识

Visual FoxPro 的远程视图可以更新远程 ODBC 数据源中的数据。在建立远程视图之前，通常先创建连接。在项目管理器中，选择某个数据库下的“连接”命令并在弹出的“连接”对话框中单击“新建”按钮，在弹出的“连接设计器连接 1”对话框中创建一个与远程数据



源的连接。

创建连接后，使用视图设计器创建远程视图的一般操作步骤如下。

在项目管理器中，选择“数据库”“新建远程视图”命令，在弹出的“新建远程视图”对话框中单击“新建视图”按钮，弹出“选择连接或数据源”对话框。

先选择“数据库中的连接”列表框中的某个选项，再单击“确定”按钮，弹出“打开”对话框。

选择作为数据源的表，再单击“添加”按钮把表添加到视图设计器，然后单击“关闭”按钮，关闭“打开”对话框。

使用创建本地视图相同的方法，在视图设计器中设计远程视图，并保存视图即可。

创建远程视图后，可以使用修改本地视图相同的方法修改远程视图，可以使用运行本地视图相同的方法运行远程视图，更新服务器上的数据。只是使用远程视图更新数据时，必须有在远程数据源写入数据的权限。



## 实施步骤

1) 在 Microsoft Access 环境下，按照表 3-2 的结构新建数据库 MyBookAccess.accdB 及其所包含的“图书信息表”(过程省略)。

2) 在 Visual FoxPro 环境下打开“学生成绩管理”数据库 StudentDB.dbc，并在该数据库中新建一个连接，具体操作步骤如下。

选择“数据库”→“连接”命令，弹出“连接”对话框，如图 3-53 所示。单击“新建”按钮，弹出如图 3-54 所示的“连接设计器连接 1”对话框。



图 3-53 “连接”对话框



图 3-54 “连接设计器连接 1”对话框

单击“新建数据源”按钮，弹出“创新数据源”对话框，选中“文件数据源(与机器无关)”单选按钮，单击“下一步”按钮，文件数据源的创建过程如图 3-55~图 3-58 所示。

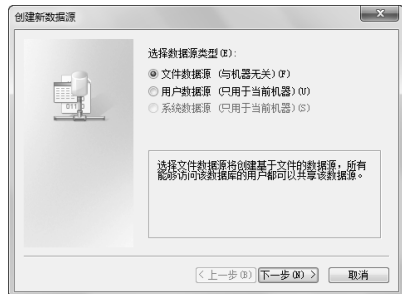


图 3-55 选择数据源类型



图 3-56 选择数据源的驱动程序



图 3-57 选择文件数据源的名称及保存路径



图 3-58 完成新建数据源

ODBC Microsoft Access 驱动程序的安装。完成新数据源的创建后，弹出“ODBC Microsoft Access 安装”对话框，按图 3-59 中所描述的步骤完成 ODBC Access 数据库的安装（ODBC 可以连接位于网络上的任意数据类型的数据库）。

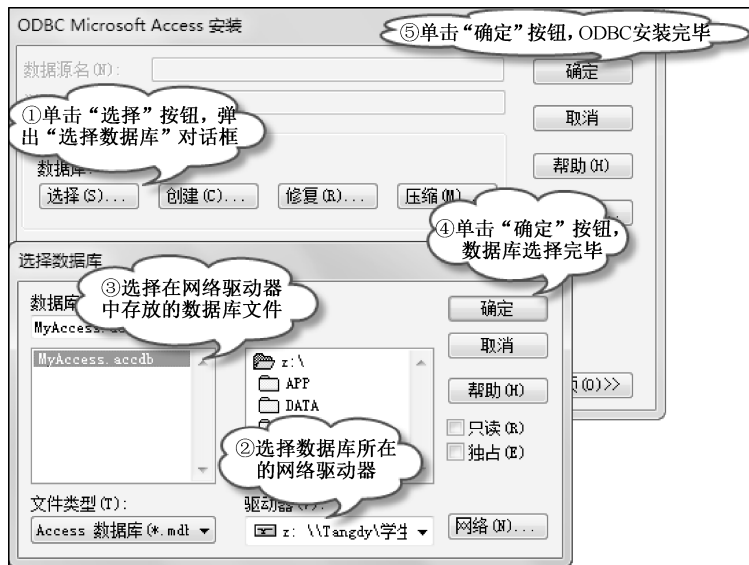


图 3-59 建立 ODBC 与数据库的连接

重返“连接设计器连接 1”对话框，继续完成对连接 1 的设置，设置步骤如图 3-60 所示。若设置正确，则在“连接串”文本框中显示连接字符串，如图 3-61 所示，单击“确定”按钮后弹出如图 3-62 所示的“连接设计器”对话框。

这样就建立好了连接 1。这个连接的作用是，让 Visual FoxPro 数据库连接到 Access 数据库，且这个连接的位置无论是本地的还是网络的，对用户来说都是透明的。

3) 下面以连接好的 Access 数据表“图书信息表”为数据源创建远程视图，其过程与创建本地视图相似。

选择“数据库”→“新建远程视图”命令，在弹出的“新建远程视图”对话框中单击“新建视图”按钮，弹出“选择连接或数据源”对话框，选择上面建好的“连接 1”选项，单击“确定”按钮，弹出“打开”对话框。

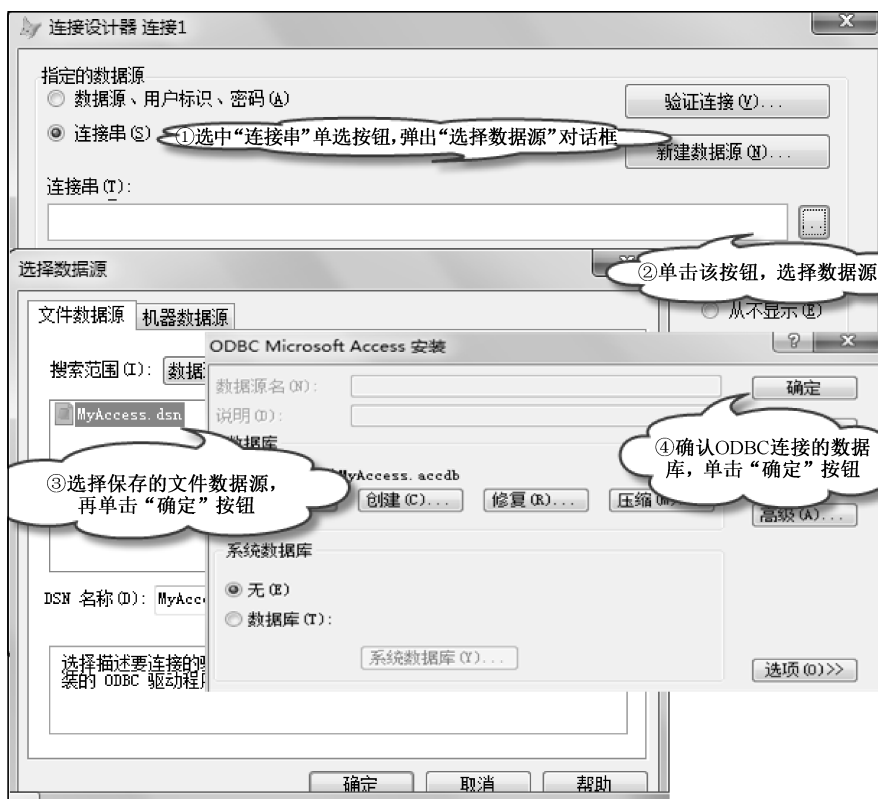


图 3-60 将连接串与数据源相关联,建立连接 1



图 3-61 自动生成与数据库相关联的连接字符串



图 3-62 “连接设计器”对话框

在“表位于连接 1”列表框中选择“table1”选项,单击“添加”按钮,如图 3-63 所示。

在视图设计器中的“字段”选项卡中将表中的字段全部添加进来,在“更新条件”选项卡中,将“图书编号”字段设置为关键字段,其他所有字段都设置为可更新字段,并勾选“Send



SQL updates”(发送 SQL 更新)复选框,如图 3-64 所示。



图 3-63 打开建立远程视图的连接数据表并命名



图 3-64 “更新条件”选项卡

运行远程视图,将图书编号“00001”的出版社由“希望出版社”修改为“机械工业出版社”,保存后关闭。再打开该远程视图连接的原数据库进行对照,数据发生了变化,如图 3-65 所示。

保存远程视图名为 RemoteAccess,如图 3-66 所示。

| 视图5 |       |            |         |         |
|-----|-------|------------|---------|---------|
|     | Id    | 图书编号       | 图书名称    | 出版社     |
| 1   | 00001 | 微机计算机原理与应用 | 希望出版社   | 希望出版社   |
| 2   | 00002 | 计算机组装与维护   | 高等教育出版社 | 高等教育出版社 |
| 3   | 00003 | 电子商务基础与实务  | 电子工业出版社 | 电子工业出版社 |

远程视图数据

| table1 |       |            |         |        |
|--------|-------|------------|---------|--------|
|        | 图书编号  | 图书名称       | 出版社     | 书价     |
| 1      | 00001 | 微机计算机原理与应用 | 机械工业出版社 | ¥24.00 |
| 2      | 00002 | 计算机组装与维护   | 高等教育出版社 | ¥24.00 |
| 3      | 00003 | 电子商务基础与实务  | 电子工业出版社 | ¥30.50 |

视图修改后Access源表

图 3-65 视图中的数据更新后与源表数据比对



图 3-66 数据库中的本地视图和远程视图

通过运行远程视图,我们可以不直接操作数据库,实现查询并更新网络中的数据。



## 阅读材料

### 1. SQL 语言

SQL (Structured Query Language, 结构化查询语言) 是用户操作关系数据库的通用语言。SQL 虽然称为结构化查询语言,而且查询操作确实是数据库中的主要操作,但并不是说 SQL 只支持查询操作,它实际上包含数据定义、数据操纵和数据控制等与数据库有关的全部功能。

SQL 已经成为关系数据库的标准语言,所以现在所有的关系数据库管理系统,包括小型数据库管理系统都支持 SQL 语言,只是不同的系统支持的 SQL 语言功能有所区别。

### 2. SQL 语言的发展

最早的 SQL 原型是 IBM 的研究人员在 20 世纪 70 年代开发的。随着 SQL 语言的颁布,各

数据库厂商纷纷在其产品中引入并支持 SQL 语言。尽管绝大多数产品对 SQL 语言的支持大部分是相似的，但它们之间也存在着一定的差异。

1986 年 10 月，美国 ANSI 采用 SQL 作为关系数据库管理系统的标准语言。随后，ISO 于 1987 年 6 月也正式采纳它为国际标准，并在此基础上进行了补充。到 1989 年 4 月，ISO 提出了具有完整性特征的 SQL，并称之为 SQL-89。SQL-89 标准的颁布，对数据库技术的发展和数据库的应用都起了很大的推动作用。尽管如此，SQL-89 仍有许多不足或不能满足应用需求的地方。为此，在 SQL-89 的基础上，经过 3 年多的研究和修改，ISO 和 ANSI 共同于 1992 年 8 月又颁布了 SQL 的新标准，即 SQL-92（或称为 SQL2）。SQL-92 标准也不是非常完善，于是在 1999 年又颁布了新的 SQL 标准，称为 SQL-99 或 SQL3。

### 3. SQL 语言的特点

SQL 是一种说明性语言。从程序员的观点来说，编程人员不必指明为得到特定结果需要计算机执行的所有操作，而只需要向数据库管理系统指明要完成的任务，然后由系统自己决定如何获得期望的结果。

SQL 不要求用户指定对数据的存放方法，这种特性使用户更易集中精力于要得到的结果。所有 SQL 语句都可以使用查询优化器，由它决定对指定数据以最快速度存取的手段。查询优化器知道存在什么索引，在哪儿使用更合适，而用户不需要知道表是否有索引，表有什么类型的索引。

SQL 语言可以交互式使用，也可以嵌入式使用。交互 SQL 允许用户直接传送命令到数据库管理系统中，且得到返回结果。嵌入式 SQL 语句则作为通用语言，作为 C、C++ 或 Java 程序代码的一部分。在这种情况下，通用编程语言是主语言，当主语言中嵌入使用 SQL 语句时，用户不用直接观察到各条 SQL 语句的执行结果，其结果是通过变量或过程参数返回的。

### 4. SQL 语言的类型

构成 SQL 语言的语句和命令一般分为两类：数据定义语言 DDL 和数据操作语言 DML。DDL 包括定义或创建数据库对象的语句，如创建表、索引和视图等。最常使用的 DDL 语句是 CREATE、ALTER 和 DROP，通过语句所带选项不同来完成对不同对象的操作。DML 包括允许对数据库对象进行处理和操作的语句，最常使用的 DML 语句是 SELECT、INSERT、DELETE 和 UPDATE。



## 项目小结

在数据库中，表是数据库的基础，而使用视图可以很好地组织表中的数据，使表中的数据可以按照用户的意图输出。本项目介绍了如何使用视图设计器创建视图及如何更新视图数据。通过本项目的学习，读者应该能够设计出所需的简单视图。



## 上机实习

1. 利用单元二建立的表 Book.dbf、Loander.dbf 和 Loand.dbf，建立满足下列要求的视图。  
查询读者的姓名、编号、电话、单位及证件号码。  
查询图书的书名、数量和作者。  
查询读者编号为 × × × × 借阅的图书编号、书名。
2. 更新 Loand 表中图书编号为 × × × × 的借阅时间。



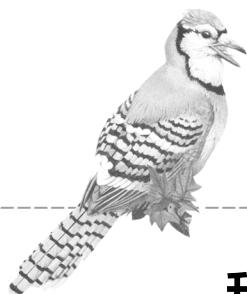
## 知识巩固

### 一、选择题

1. 视图设计器比查询设计器多出的选项卡是 ( )。  
A. 字段                  B. 排序                  C. 联接                  D. 更新条件
2. 下列关于视图的叙述中, 正确的是 ( )。  
A. 只能由自由表创建视图                  B. 只能由数据库表创建视图  
C. 不能由自由表创建视图                  D. 自由表和数据库表都能创建视图
3. 视图是一个 ( )。  
A. 虚拟的表                  B. 真实的表  
C. 不依赖于数据库的表                  D. 不能修改的表
4. 关于查询与视图的错误叙述是 ( )。  
A. 查询和视图都可以从一个或多个表中提取数据  
B. 查询是作为文本文件, 以扩展名.qpr 存储的  
C. 可以通过视图更改源表的数据  
D. 视图是完全独立的, 它不依赖于数据库的存在而存在
5. 视图不能单独存在, 它必须依赖于 ( )。  
A. 视图                  B. 数据库                  C. 数据表                  D. 查询

### 二、填空题

1. 在项目管理器中运行视图时, 先选择\_\_\_\_\_, 再单击\_\_\_\_\_按钮。
2. 在项目管理器中先选择视图, 再单击\_\_\_\_\_按钮, 可以修改选定的视图。
3. 本地视图用于\_\_\_\_\_, 远程视图用于\_\_\_\_\_。
4. 视图设计器中的“更新条件”选项卡用于\_\_\_\_\_。



# 单元四

## 程序设计基础

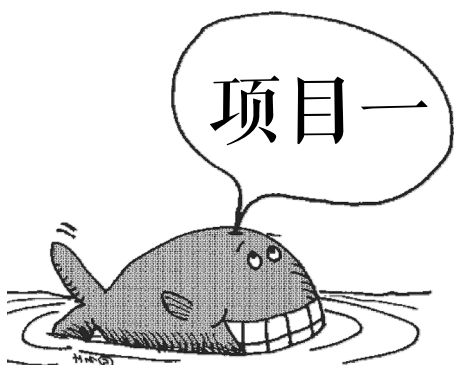
### 知识目标

- 1) 了解结构化程序设计的基本思想。
- 2) 熟练掌握创建、运行、调试程序文件的方法。
- 3) 熟练掌握常用输入、输出函数的使用方法。
- 4) 掌握顺序结构程序设计的思路和方法。
- 5) 掌握选择结构程序设计的思路和方法。
- 6) 掌握循环结构程序设计的思路和方法。
- 7) 掌握多模块程序设计的基础知识。

### 技能目标

- 1) 能够熟练创建、编辑和运行程序。
- 2) 能够进行简单的程序编译与调试。
- 3) 能够熟练应用输入、输出函数。
- 4) 能够熟练应用 3 种基本结构设计程序。
- 5) 能够应用多模块设计思想编写程序，降低程序的复杂性。





# 编写程序输出指定学号的学生信息

## 项目说明

Visual FoxPro 9.0 系统提供了一套功能完善的程序语言系统，即面向过程的结构化程序设计和面向对象的可视化程序设计。在实际应用中，程序方式是数据库管理的另外一种操作方式，也是最常用的工作方式。本单元主要介绍面向过程的结构化程序设计。

程序是为了完成某一具体任务而编写的一系列指令。结构化程序设计就是将若干条命令或语句按一定的逻辑结构组织编写的过程。程序设计是把一组 Visual FoxPro 命令按照所开发的软件要求组织在程序文件中，运行程序文件即可自动执行其中的命令，完成数据处理功能。Visual FoxPro 中的程序结构可以分为顺序结构、选择结构和循环结构。

顺序结构是最简单、最常用的基本程序结构，它按照命令或语句的排列顺序依次执行，直至将程序中的每条命令或语句都执行完成。在编程中，分支问题应该应用选择结构来解决，即根据条件选择执行不同的程序分支。当需要对数据表中的多条记录进行相同的操作时，为了避免编程时代码冗余，就需要用到循环结构。循环结构用于在指定的条件下反复执行某些相同的操作。

本项目主要介绍在 Visual FoxPro 系统中创建、调试和运行程序的方法；常用输入、输出函数的应用及顺序、选择、循环 3 种基本程序结构的设计和应用。本项目包括 6 个任务。

任务一：创建、调试并运行程序。

任务二：从键盘输入学号，查询并显示。

任务三：格式化输出学生信息。

任务四：如果学号不存在，输出提示信息。

任务五：判断该学生成绩，并格式化输出成绩等级。

任务六：格式化输出所有学生的信息。

## 任务一 创建、调试并运行程序

### 任务目标

能够使用命令或菜单熟练创建程序文件。



掌握编写程序文件的基本原则。  
能够使用命令或菜单编译并运行程序。  
能够进行简单的程序调试。

## 任务描述

新建程序文件，打开学生自然表 Student.dbf，查询并显示学号为“120104”的学生记录。

## 相关知识

程序文件是文本文件，用户可以用任何文字处理软件创建程序文件，也可以用 Visual FoxPro 中提供的菜单或命令创建程序文件。程序文件的默认扩展名是“.prg”。

在 Visual FoxPro 系统中，组成程序文件的大多数语句其实就是在“命令”对话框中输入的各种命令。根据应用中的具体需要，将这些命令按照一定的顺序及逻辑关系组织起来放在一个文件中。当需要执行这些命令时，直接调用程序文件，系统就会自动批处理执行程序文件中的命令。即使对程序设计并不了解的用户，也可以通过调用程序文件而执行其中的命令。

### 1. 创建程序文件

创建程序文件既可以通过菜单方式，也可以通过命令方式。多数时候是通过命令方式建立的。

#### (1) 命令方式

命令格式：

```
MODIFY COMMAND 程序文件名 | ?
```

功能：新建或打开一个已有的程序文件。

说明：

- 程序文件名 是要创建或打开的文件名，当该项省略时，新建默认文件名为“程序 1”的程序文件。
- 程序文件的默认扩展名为“.prg”。
- 如果命令后面是“？”，则弹出“打开”对话框，可以选择要打开的文件。
- 若磁盘上已有该文件，则此命令的作用是打开程序文件，可以对该文件进行编辑、修改；若无此文件，则此命令将建立一个新的程序文件。
- 文件编辑好后，可按 Ctrl+W 组合键或选择“文件”→“保存”命令保存程序，返回“命令”对话框中，或按 Esc 键放弃存盘并退出。
- 刚刚建立的程序文件，如果希望它存放在指定的文件夹下，可以在程序文件名前加上路径信息，如输入“MODIFY COMMAND D:\VFP9\abc”。没有指明保存路径的程序文件一般被保存在系统默认的文件夹中。

#### (2) 菜单方式

选择“文件”→“新建”命令，在弹出的如图 4-1 所示的“新建”对话框中选中“程序”单选按钮，再单击“新建文件”按钮，打开如图 4-2 所示的“程序 1”窗口，在其中输入程序命令序列。

通常用菜单建立新文件时，由于用户事先没有给该文件命名，因此首次将该文件存盘时，Visual FoxPro 将自动弹出如图 4-3 所示的“另存为”对话框，提示用户给程序文件命名。



图 4-1 “新建”对话框

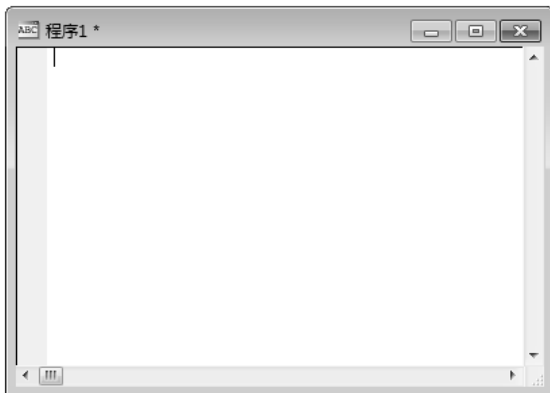


图 4-2 “程序 1”窗口

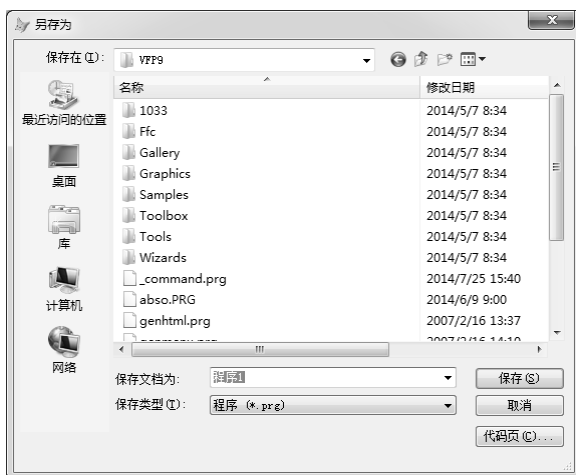


图 4-3 “另存为”对话框

## 2. 程序设计的基本原则

编写程序时，必须遵循 Visual FoxPro 程序的一些基本规则。

- 一个程序文件由若干行组成。
- 每一行由一条语句或一条命令组成。
- 每一行都以 Enter 键结束，若一行写不完一条命令，可在分行处加上续行符号“;”，在下一行继续编写。
- 在程序末尾一般添加结束语句，如 RETURN、QUIT 等，表示程序结束。
- 为了便于阅读程序，同级的语句左对齐，不同级别的语句用缩进方式进行区分。
- 在程序中任意空行处或任意行的后面可使用注释语句，对程序或该程序行进行说明，以增加程序的易读性。注释语句的内容由“\*”或 NOTE 命令来引导。对程序进行注释一般是在程序行后用“&&”引导注释内容，注释语句和内容对于程序的运行无任何影响。

## 3. 程序编辑窗口

程序编辑窗口的快捷菜单提供了对当前程序和程序编辑窗口可执行的各种命令，如图 4-4 所示。这些命令主要分为两类：一类是执行和调试程序的命令，如执行、设置断点等；另一类

是设置编辑窗口格式的命令，如字体、属性等。

快捷菜单中的“属性”命令提供了对当前程序编辑窗口操作的各种选择，如动作、外观、保存选项等，如图 4-5 所示。



图 4-4 程序编辑窗口的快捷菜单



图 4-5 “属性”对话框

程序编辑窗口根据命令的类别不同显示不同的颜色，这可以选择“工具”“选项”命令，弹出“选项”对话框，在“编辑器”选项卡中的“语法着色”选项组中进行修改，如图 4-6 所示。

在编辑程序的过程中，可以同时打开多个文本编辑器窗口编辑多个文件，可以在一个文件或多个文件之间对文本进行剪切、复制或粘贴等操作。

#### 4. 运行程序文件

程序文件创建后，要运行该程序，一般可使用菜单方式或命令方式。

##### (1) 菜单方式

选择“程序”→“运行”命令，弹出“运行”对话框，如图 4-7 所示。在该对话框中选择要运行的程序文件名，单击“运行”按钮即可。程序的运行结果将在工作区窗口中显示。



图 4-6 “编辑器”选项卡的设置

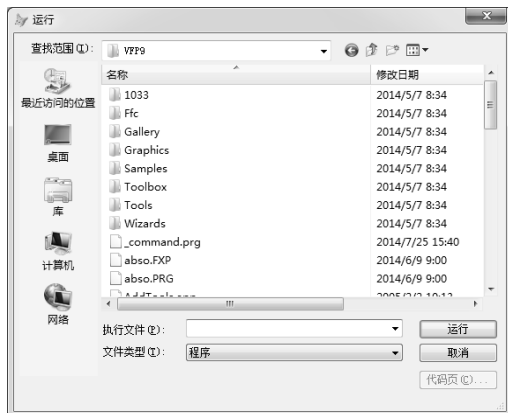


图 4-7 在“运行”对话框中选择运行的程序



## (2) 命令方式

使用 DO 命令可以运行一个程序文件，其命令格式如下：

DO <程序文件名>

功能：运行指定的程序文件。

说明：

- 在“命令”对话框中输入此命令，即可将指定的文件调入内存并运行。
- 运行此命令时，如果要执行的程序不在默认路径下，必须指明文件所在的路径。
- DO 命令默认的是运行程序文件，如果后面的文件名没带扩展名，则系统默认为“.prg”。
- DO 命令不仅可以运行程序文件，也可以运行查询和表单等。

运行程序后，系统自动将“.prg”文件编译成“.fxp”文件，运行结果显示在系统主窗口中。如果程序中存在错误，在运行程序过程中，Visual FoxPro 会提示程序中存在的错误，弹出如图 4-8 所示的“程序错误”对话框，给出错误原因。其中“取消”表示终止程序运行，返回“命令”对话框；“挂起”表示暂停程序；“忽略”表示跳过出错的语句继续执行；“帮助”表示要求系统对于错误做更详细的说明。

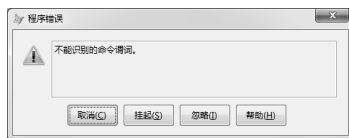


图 4-8 “程序错误”对话框



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行，新建程序文件。

MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4\_1

如图 4-9 所示，在程序编辑窗口输入代码。

按 Ctrl+W 键保存程序文件。

运行程序文件“D:\学生成绩管理信息系统\APP\ chengxu4\_1.prg”，在“命令”对话框中输入以下命令：

DO D:\学生成绩管理信息系统\APP\ chengxu4\_1

查看程序运行结果。学号为“120104”的学生记录显示在 Visual FoxPro 的工作区窗口中，如图 4-10 所示。

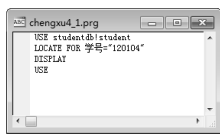


图 4-9 程序编辑窗口



图 4-10 程序运行结果



## 问题探究

1. 如何在项目管理器中创建、运行、修改程序文件？

### (1) 创建程序文件

打开“学生成绩管理信息系统”项目文件，如图 4-11 所示，在项目管理器的“代码”选项卡中，选择“程序”选项，单击“新建”按钮，打开编辑程序文件的程序编辑窗口。在窗口中输入程序命令序列，保存程序文件。

### (2) 运行程序文件

在“学生成绩管理信息系统”项目管理器的“代码”选项卡中,选择需要运行的程序文件,单击“运行”按钮运行程序。

### (3) 修改程序文件

在“学生成绩管理信息系统”项目管理器的“代码”选项卡中,选择需要修改的程序文件,单击“修改”按钮打开程序文件。修改程序文件,最后保存退出。



图 4-11 “学生成绩管理信息系统”项目管理器

## 任务二 从键盘输入学号, 查询并显示

### 任务目标

能够熟练应用 3 种交互式键盘输入命令。

### 任务描述

编写程序文件, 请用户从键盘输入学号, 在表 Student.dbf 中查询并显示该学生的信息。

### 相关知识

在程序运行过程中, 往往需要人工从键盘输入数据。例如, 查询某个学生的自然信息, 需要输入学生的学号或姓名, 才会输出相应学生的信息。Visual FoxPro 提供了简单的交互式输入命令 WAIT、ACCEPT 和 INPUT。

#### 1. 单字符输入命令 WAIT

命令格式:

WAIT [ 提示信息 ] [ TO 内存变量 ] [ TIMEOUT 数值表达式 ]

功能: 暂停程序的运行, 等待用户从键盘上输入单个字符给指定的内存变量。

参数:

- [ 提示信息 ] 是在屏幕上显示提示用户进行操作的信息。提示信息 必须用字符型定界符括起来。省略该选项时, 则提示“按任意键继续……”, 并等待用户按键操作。

- [ TO 内存变量 ] 是可选项。当选用时, 从键盘输入的字符被保存到指定的内存变量中。省略该选项时, 则不保存输入的字符。

- 该命令只能接受一个字符, 在输入时不需要用定界符, 也不需要按 Enter 键。

- [ TIMEOUT 数值表达式 ] 用来设定等待的时间(秒)。如果用户超过等待时间仍没有按键, 则系统自动继续执行。

#### 2. 表达式输入命令 ACCEPT

命令格式:

ACCEPT [ 提示信息 ] TO 内存变量



功能:暂停程序的运行,等待用户从键盘上输入字符串,存入指定的内存变量中,按 Enter 键可结束输入,继续执行后面的命令。

说明:

- [ 提示信息 ] 用于在屏幕上显示提示用户进行的操作的信息。提示信息 是字符型表达式,必须用字符型定界符括起来。省略该选项时,则不显示任何提示信息。

- 用户在输入时,无论输入的是中文、英文、数字还是表达式,都当作字符处理,即自动在输入的数据两侧加上字符型数据的定界符。输入字符串的长度不能超过 254 个字符 (127 个汉字)。

### 3. 字符输入命令 INPUT

命令格式:

INPUT [ 提示信息 ] TO 内存变量

功能:该命令接受从键盘上输入的数据并赋值给指定的内存变量,按 Enter 键可结束输入,继续执行后面的命令。

参数:

- [ 提示信息 ] 用于在屏幕上显示信息,提示用户进行下一步操作。
- 内存变量 的类型取决于输入数据的类型。

该命令接收的数据类型可以是字符型、数值型、逻辑型或日期型。数值型数据可以直接输入;字符型数据输入时必须使用定界符括起来;逻辑型数据两边要加圆点,即“.F.”或“.T.”;日期型常量要用花括号,如{^2014-06-05},或使用转换函数进行转换,如使用 CTOD()转换函数或 DATA()函数作为输入的数据。



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行,新建程序文件。

MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4\_2

在程序编辑窗口输入以下代码:

```
USE StudentDB!Student
ACCEPT "请输入学生学号" TO xh
LOCATE FOR 学号=xh
DISPLAY
USE
```

按 Ctrl+W 组合键保存程序文件。

运行程序文件 chengxu4\_2,在“命令”对话框中输入以下命令:

DO D:\学生成绩管理信息系统\APP\ chengxu4\_2

查看程序运行结果。根据用户输入的学号,相应的学生记录显示在 Visual FoxPro 的工作区窗口中,如图 4-12 所示。

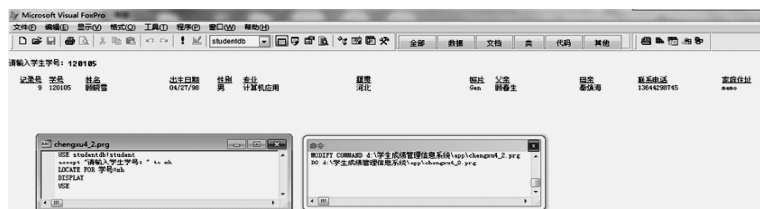


图 4-12 从键盘输入学号查询结果



## 问题探究

在实际应用中，如何区别使用 WAIT、ACCEPT 和 INPUT 三种输入命令？

WAIT、ACCEPT 和 INPUT 命令都用于与用户的交互输入，也都能够输出显示信息，它们之间存在如下区别。

WAIT 命令只能输入一个字符，ACCEPT 命令输入的是字符串，INPUT 命令可以将输入的表达式值保存到内存变量中。因此，WAIT 和 ACCEPT 命令中的变量一定是字符型变量，而 INPUT 命令中的变量类型由输入表达式值的数据类型决定。

使用 WAIT 命令输入数据时，只需输入一个字符，不需按 Enter 键。而使用 ACCEPT 和 INPUT 命令输入数据时，要按 Enter 键表示输入结束。

使用 WAIT 和 ACCEPT 命令输入字符型数据时不用加定界符。而使用 INPUT 命令输入字符串则必须加定界符。

ACCEPT 命令和 INPUT 命令只能在屏幕的当前光标下一行开始显示数据，而 WAIT 命令既可以在屏幕上显示数据，也可以在系统窗口中显示数据。

综上所述，WAIT 命令常用于输入单字符和显示信息，ACCEPT 命令常用于输入字符串，INPUT 命令常用于输入其他类型的数据。

## 任务三 格式化输出学生信息

### 任务目标

理解顺序结构程序设计思想。

能够阅读、设计、编写顺序结构程序。

熟练应用格式化输出命令。

会应用程序设计中的辅助命令。

### 任务描述

编写程序文件，请用户从键盘输入需查询学生的学号，在表 Student.dbf 中查询并格式化输出该学生的信息。

### 相关知识

#### 1. 顺序结构程序设计

顺序结构的程序按从上到下的顺序执行语句。它是最基本的程序结构，也是适用最广泛的程序结构。设计顺序结构程序时，首先仔细分析需要解决的问题，找出解决问题的方法和操作步骤，再根据操作步骤写出对应的命令，将这些命令按照操作步骤的顺序组织在程序文件中，最后让计算机自动执行，以完成指定的任务。

#### 2. 程序设计中的常用命令

##### (1) 清屏命令 (CLEAR)



### 命令格式：

CLEAR

功能：清除当前主窗口上的所有信息。

### (2) 清除内存变量命令 (CLEAR ALL)

### 命令格式：

CLEAR ALL

功能：关闭打开的数据库、表及索引文件，清除用户自定义的内存变量和对象。

### (3) 非格式输出命令 ( ? / ? ? )

一个独立的程序中可以没有任何输入语句，但应该至少有一条输出语句来显示程序执行的结果。

### 命令格式：

? 表达式列表

? ? 表达式列表

功能：显示常量、内存变量或表达式的值。

### 参数：

● ? 表达式列表：命令在执行时，先换行，计算表达式值，将值输出。

● ? ? 表达式列表：命令是从当前光标位置开始输出表达式的值。

此命令一次可以同时输出多个表达式的值，多个表达式间可用逗号分割。

### (4) 格式输出命令 (@ ..... SAY .....)

### 命令格式：

@ 行号，列号 SAY 表达式

功能：在指定的屏幕坐标位置显示输出表达式的值。

### 参数：

● 行号，列号 是一对数值表达式，确定输出起始位置的屏幕坐标。

● 表达式 的内容可以是数值、字符、内存变量和字段变量，是输出的数据。



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行，新建程序文件。

MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4\_3

在程序编辑窗口输入以下代码：

```
CLEAR
USE StudentDB!Student
ACCEPT "请输入学生学号：" TO xh
LOCATE FOR 学号=xh
@ 3,45 SAY "学 生 信 息"
@ 8,25 SAY "学号：" +学号
@ 8,55 SAY 照片
@ 10,25 SAY "姓名：" +姓名
@ 12,25 SAY "性别：" +性别
@ 14,25 SAY "出生日期：" +DTOC(出生日期)
@ 16,25 SAY "籍贯：" +籍贯
@ 18,25 SAY "专业：" +专业
USE
```

按 Ctrl+W 组合键保存程序文件。



运行程序文件 chengxu4\_3，在“命令”对话框中输入以下命令：

```
DO D:\学生成绩管理信息系统\APP\ chengxu4_3
```

查看程序运行结果。根据用户输入的学号，相应的学生信息格式化显示在 Visual FoxPro 的工作区窗口中，如图 4-13 所示。

请输入学生学号:120102

学 生 信 息

学号: 120102

姓名: 王晓明

性别: 女

出生日期: 11/07/98

籍贯: 黑龙江

专业: 计算机应用



图 4-13 格式化输出命令运行结果

## 任务四 如果学号不存在，输出提示信息

### 任务目标

理解分支结构程序设计思想。

能够阅读、设计、编写简单分支结构程序。

熟练应用 IF...ELSE...ENDIF 语句。

### 任务描述

编写程序文件，请用户从键盘输入需查询学生的学号，在表 Student.dbf 中查询。如果表中不存在该学号的记录，输出提示信息“抱歉，该学号不存在！”；如果查询到该学号，格式化输出该学生的信息。

### 相关知识

程序在运行时，一般情况按照语句在文件中的先后顺序执行。但在实际数据处理过程中，常常需要根据是否满足一定的条件来决定下面要执行什么语句。例如，人事部门希望把职称为工程师的人员名单打印出来。这样，就需要在程序中设置一个条件“是否为工程师”，然后在程序执行时判断是否满足条件。如果满足条件“是工程师”，则执行“打印名单”的语句；如果不满足条件即“不是工程师”，则执行另一些语句（在此处退出执行程序）。这类问题在程序设计上称为选择结构，应用分支语句来实现不同条件的选择。

#### 1. 单分支结构

单分支结构又称为“简单分支”，即只有一个程序分支，根据用户设置的条件表达式的值，决定是否执行该程序分支的语句。单分支选择结构 IF...ENDIF 语句的格式如下：

```
IF 条件表达式
 语句序列
ENDIF
```

功能：首先计算 条件表达式 的值，当表达式的值为真时，顺序执行 IF 和 ENDIF 之间的 语句序列，然后再执行 ENDIF 后面的语句；当表达式的值为假时，直接跳到 ENDIF 后面的语句继续执行。



说明：

● IF 表示选择结构开始，ENDIF 表示选择结构结束，它们必须成对使用，缺一不可。

● 语句序列 由一组 Visual FoxPro 的程序语句组成。

IF...ENDIF 语句的执行过程如图 4-14 所示。

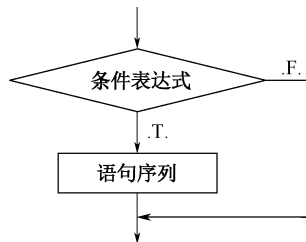


图 4-14 IF...ENDIF 语句执行流程

## 2. 双分支结构

双分支选择结构有两个程序分支。无论是否满足指定的条件，都能执行其中一个程序分支，并且只能执行一个程序分支。双分支选择结构 IF...ELSE...ENDIF 语句的格式如下：

```

IF 条件表达式
 语句序列 1
ELSE
 语句序列 2
ENDIF

```

功能：执行此语句，计算机首先判断条件表达式的值，若为真，执行 语句序列 1，然后跳过 语句序列 2，从 ENDIF 后面的语句开始继续执行；若条件表达式的值为假，则直接执行 语句序列 2，然后再继续执行 ENDIF 后面的语句。

说明：

● IF...ELSE...ENDIF 语句必须成对使用。

● 省略 ELSE 选项，该结构就成为单分支选择结构。

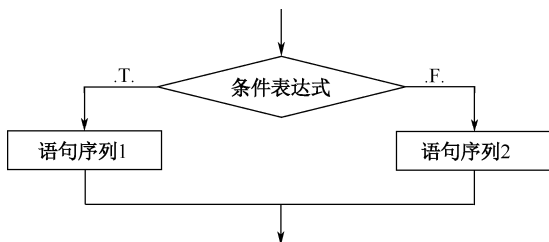


图 4-15 双分支结构的执行过程

● 语句序列 1 和 语句序列 2 中可以再嵌套 IF 命令。但要注意 IF 语句的出口 ENDIF 的个数及它所在的位置，IF 和 ENDIF 必须一一对应，外层的 IF 与 ENDIF 一定包含内层的 IF 与 ENDIF，不能交叉。另外，为使 IF 语句嵌套结构清晰，最好采用缩进格式书写程序。

IF...ELSE...ENDIF 语句的执行过程如图 4-15 所示。



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行，新建程序文件。

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4_4
```

在程序编辑窗口输入以下代码：

```

CLEAR
USE StudentDB!Student
ACCEPT "请输入学生学号" TO xh
LOCATE FOR 学号=xh
IF Found()
@ 3,45 SAY "学 生 信 息"
@ 8,25 SAY "学号：" +学号

```

```

@ 8,55 SAY 照片
@ 10,25 SAY " 姓名： " +姓名
@ 12,25 SAY " 性别： " +性别
@ 14,25 SAY " 出生日期： " +DTOC(出生日期)
@ 16,25 SAY " 籍贯： " +籍贯
@ 18,25 SAY " 专业： " +专业
ELSE
? " 抱歉，该学号不存在！ "
ENDIF
USE

```

按 Ctrl+W 组合键保存程序文件。

运行程序文件 chengxu4\_4，在“命令”对话框中输入以下命令：

```
DO D:\学生成绩管理信息系统\APP\ chengxu4_4
```

查看程序运行结果。如果用户输入的学号查询不到，显示提示信息，如图 4-16 所示；如果用户输入的学号能查询到，运行结果格式参照 chengxu4\_3。



## 问题探究

如何应用 IF...ENDIF 语句实现多分支选择结构？

可以采用嵌套条件语句来实现多分支结构，所谓嵌套条件语句就是将一个 IF...ENDIF 程序结构设置在另一个 IF...ENDIF 程序结构内，一层套着一层，以满足超过两个条件需要进行选择的情况。

嵌套式 IF 条件语句的使用方法可以参考如下代码：

```

CLEAR
USE StudentDB!Student
ACCEPT " 请输入学生学号 " TO xh
LOCATE FOR 学号=xh
IF Found()
 IF 籍贯=" 辽宁 "
 DISPLAY
 ELSE
 ? " 该学生的籍贯不是辽宁 "
 ENDIF
ELSE
 ? " 抱歉，该学号不存在！ "
ENDIF
USE

```

在本程序中，使用两层 IF...ENDIF 命令语句嵌套，外层条件语句用来判断学号是否存在，如果不存在，输出提示信息；内层条件语句判断该学生的籍贯是否为辽宁，然后根据条件执行不同的操作。

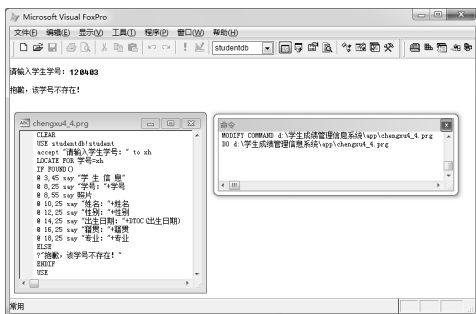


图 4-16 分支结构程序的运行结果

## 任务五 判断学生成绩，并格式化输出成绩等级

### 任务目标

理解多分支选择结构程序的设计思想。



熟练应用 DO CASE...ENDCASE 语句。  
能够应用分支语句的嵌套，设计编写复杂分支结构程序。

## 任务描述

编写程序文件，请用户从键盘输入需查询的学生学号，查询该学生信息。如果表中存在该学号，那么格式化输出该学生信息，并且根据成绩输出“优、良、中、及格、不及格”的成绩等级；否则，输出提示信息“该学号不存在！”。

其中，成绩等级的范围规定如下：优为大于等于 90；良为大于等于 80 且小于 90；中为大于等于 70 且小于 80；及格为大于等于 60 且小于 70；不及格为小于 60。

## 相关知识

在 IF 选择语句中，根据一个条件，产生两种选择。但在实际生活中，常常会有多种可供选择的问题。虽然用 IF 语句嵌套的方式也可以解决多重选择的问题，但是，当提供选择的方案较多时，用 IF 语句嵌套既复杂又容易出错。所以，多分支问题最好用多分支选择结构来解决。

多分支选择结构通常有多个程序分支，但最多执行其中一个程序分支，也可能所有程序分支都不执行。多分支选择结构 DO CASE...ENDCASE 的格式如下。

```
DO CASE
CASE 条件表达式 1
语句序列 1
CASE 条件表达式 2
语句序列 2
.....
CASE 条件表达式 N
语句序列 N
[OTHERWISE
语句序列 N+1]
ENDCASE
```

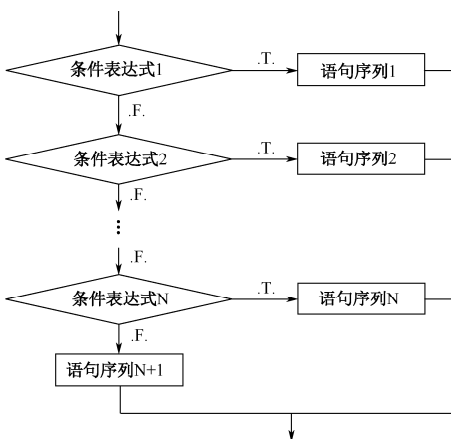


图 4-17 多分支结构的执行过程

说明：

功能：执行该语句时，系统依次查看每个 CASE 后的条件表达式的值，碰到第一个条件成立者，系统则执行相应后面的语句行序列（语句行序列可由一条或多条语句组成）；执行完此语句行序列后，跳过其他条件下的语句，直接执行 ENDCASE 后面的语句。如果所有 CASE 后的条件都不成立，在有选项 OTHERWISE 的情况下，则执行 OTHERWISE 后面的语句序列，执行完后再接着执行 ENDCASE 后面的语句；在没有可选项 OTHERWISE 的情况下，则不执行 DO CASE...ENDCASE 中的任何语句，而继续执行 ENDCASE 后面的语句。

DO CASE...ENDCASE 语句的执行过程如图 4-17 所示。

- DO CASE 和 ENDCASE 必须成对出现。
- 在 DO CASE 语句中，如果条件为真的情况多于一个，则只执行第一个条件为真者。
- DO CASE...ENDCASE 语句和 IF...ENDIF 语句可以互相嵌套，但它们之间的层次关系不能乱。



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行，新建程序文件。

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4_5
```

在程序编辑窗口输入以下代码：

```
USE D:\ 学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\ 学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
ACCEPT " 请输入学生学号：" TO xh
LOCATE FOR 学号=xh
IF Found()
 SELECT SelCourse
 LOCATE FOR 学号=xh
 cj=" "
 IF Found()
 DO CASE
 CASE 总成绩>=90 AND 总成绩<=100
 cj=" 优 "
 CASE 总成绩>=80 AND 总成绩<90
 cj=" 良 "
 CASE 总成绩>=70 AND 总成绩<80
 cj=" 中 "
 CASE 总成绩>=60 AND 总成绩<70
 cj=" 及格 "
 CASE 总成绩< 60
 cj=" 不及格 "
 ENDCASE
 ENDIF
 SELECT student
 @10,10 SAY " 学号：" +学号
 @12,10 SAY " 姓名：" +姓名
 @14,10 SAY " 性别：" +性别
 @16,10 SAY " 专业：" +专业
 @18,10 SAY " 成绩：" +cj
ELSE
 ? " 该学号不存在！ "
```



```
ENDIF
```

```
CLOSE TABLES ALL
```

按 Ctrl+W 组合键保存程序文件。

运行程序文件 chengxu4\_5，在“命令”对话框中输入以下命令：

```
DO D:\学生成绩管理信息系统\APP\chengxu4_5
```

查看程序运行结果。如果用户输入的学号能查询到，运行结果如图 4-18 所示；如果用户输入的学号查询不到，运行结果参照图 4-16。

## 任务六 格式化输出所有学生的信息

### 任务目标

理解循环结构程序的设计思想。

熟练应用 DO WHILE...ENDDO、SCAN...ENDSCAN 和 FOR...ENDFOR 这 3 种循环语句。

能够综合应用 3 种基本程序结构，设计编写程序。

### 任务描述

格式化输出学生自然表 Student.dbf 中所有学生的自然信息及成绩信息。

### 相关知识

循环结构由指定的条件控制来反复执行某些相同的操作，被反复执行的操作称为循环体。实现循环操作的程序称为循环结构程序。与顺序结构、选择结构的不同之处主要表现在：组成循环结构的循环体部分可以被反复执行，而组成顺序结构或选择结构程序中的每一条语句，最多执行一次。

Visual FoxPro 提供了基于条件、针对记录和固定次数 3 种循环结构。

#### 1. 基于条件的循环结构

DO WHILE...ENDDO 循环称为条件循环，即根据条件表达式的值，决定是否执行循环体内的语句。该循环结构语句的格式如下：

```
DO WHILE 条件表达式
 语句序列 1
[LOOP]
 语句序列 2
[EXIT]
 语句序列 3
ENDDO
```

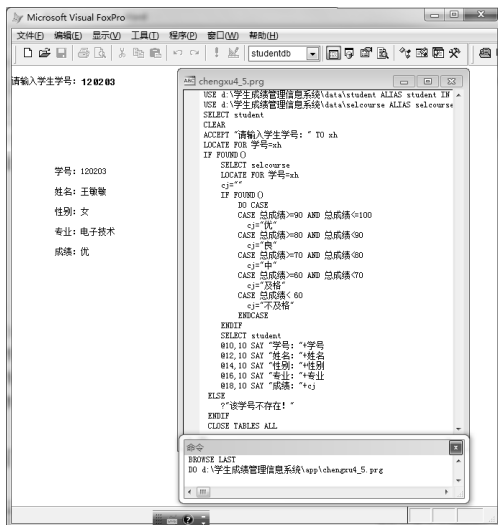


图 4-18 多分支程序结构的运行结果

功能：首先计算循环起始语句（DO WHILE）中的 条件表达式 的值，当其值为真时，执行循环体中的语句，遇到循环终止语句（ENDDO）或遇到 LOOP 就返回 DO WHILE，重新判断 条件表达式 的值；若其值仍为真，则重复上述操作，直至其值为假（或遇到 EXIT 语句）时，退出循环而执行循环终止语句（ENDDO）后面的语句。

说明：

- DO WHILE 和 ENDDO 必须成对出现，DO WHILE 是循环的入口，而 ENDDO 是循环的终止语句，中间的 语句序列 则是重复执行的循环体。

- LOOP 和 EXIT 只能在循环体中使用，可以放在循环体中的任意位置，其中 LOOP 是中止本次循环转至循环的入口处，而 EXIT 是强行退出循环。通常把它们安排在循环体的选择结构中，实现满足某种条件就终止本次循环或结束循环的功能。

例如，下列程序段判断一个自然数是否为质数，其中 LOOP 和 EXIT 的用法如下：

```
CLEAR
INPUT " 请输入一个大于 1 的自然数： " TO N
J=2
DO WHILE J<N &&循环入口
IF Mod(N,J)<>0
J=J+1
LOOP &&跳出本次循环，转到循环入口
ELSE
EXIT &&退出循环体，顺序执行 ENDDO 之后的语句
ENDIF
ENDDO
IF J==N &&输出结果
?N, " 是质数 "
ELSE
?N, " 不是质数 "
ENDIF
RETURN
```

- 循环体中一定要包含修改循环条件、使循环能够结束的语句，即如果 条件表达式 的值始终为真，则会造成死循环。

- 循环结构允许嵌套。既可以在循环中嵌套循环，也可以在循环中嵌套前面所讲的分支结构。

## 2. 针对表记录的循环结构

SCAN...ENDSCAN 语句是 Visual FoxPro 提供的专门针对当前数据库表记录的循环结构，是根据用户设置的当前记录指针，对一组记录进行循环操作。该循环结构的基本格式如下：

```
SCAN[范围][FOR 条件表达式 1][WHILE 条件表达式 2]
语句序列 1
[LOOP]
语句序列 2
[EXIT]
语句序列 3
ENDSCAN
```

功能：在当前表中按条件选择记录。如果条件满足，则执行循环体，否则结束循环，执行 ENDSCAN 后面的语句。

说明：

- 循环终止语句 ENDSCAN 标明了循环程序段的结束，它必须与 SCAN 语句成对出现。



● SCAN...ENDSCAN 语句会自动移动记录指针，所以在该循环结构的循环体中通常不使用移动记录指针的命令。

● 在循环起始语句 SCAN 中，[ 范围 ]子句指明了扫描记录的范围，其默认值为 ALL。

● 如果既没有 FOR 语句，也没有 WHILE 子句，只要当前表不为空，该结构会对当前记录执行循环体的语句，再将记录指针下移一条记录，再执行循环体语句。如此循环，直到记录指针指向文件尾，才结束循环。

● FOR 子句说明只是对使 条件表达式 1 的值为真的记录进行操作。

● WHILE 子句指定只对使 条件表达式 2 的值为真的记录进行相应操作，直至遇到使其值为假的记录为止。

● LOOP 与 EXIT 的作用与在 DO WHILE 语句中的作用相同。

使用 SCAN...ENDSCAN 循环可以方便地对当前表中所有满足条件的记录进行操作，对记录的扫描操作比 DO WHILE...ENDDO 循环要简单得多。

### 3. 固定次数循环结构

在设计循环程序时，当循环体执行的次数固定时，可以使用 FOR...ENDFOR 循环，其基本格式如下：

```
FOR 循环控制变量 = 初值 TO 终值 [STEP 步长]
 语句序列 1
 [LOOP]
 语句序列 2
 [EXIT]
 语句序列 3
ENDFOR/NEXT
```

功能：执行该循环时，首先将循环初值赋给循环控制变量，然后判断循环控制变量的值是否超过终值。若超过，则跳出循环，执行 ENDFOR 或 NEXT 后面的语句，否则重复执行循环体。当遇到 ENDFOR 或 NEXT 语句时，返回 FOR 语句，并将循环控制变量与步长相加后的值再一次与循环终值比较。如此重复执行，直到循环控制变量的值超过循环终值或遇到 EXIT 语句，结束循环。

说明：

● 步长值既可为正数，也可为负数。当 STEP 步长值省略时，系统默认步长为 1。

● ENDFOR 或 NEXT 语句为循环终止语句，其作用是表明循环程序段的终点，同时使循环变量的当前值增加一个步长。它必须与 FOR 成对出现。

● LOOP 与 EXIT 语句的作用与在 DO WHILE 语句中的作用相同。

● 循环控制变量的值超过循环终值的含义是指：当步长为正值时，循环控制变量的当前值大于终值；当步长为负值时，循环控制变量的当前值小于终值。因此，在编写 FOR 语句时，若循环控制变量的步长为正，其终值应大于初值；反之若循环控制变量的步长为负，其终值应小于初值。

例如，下列程序段用于计算 1~100 之间所有偶数的和，其中 FOR 循环的用法如下：

```
CLEAR
SUM=0
FOR N=1 TO 100 STEP 1
 IF N%2==0
 SUM=SUM+N
```



```

ENDIF
ENDFOR
?"1~100 之间所有偶数的和为：SUM=", SUM
RETURN

```

运行结果：

1~100之间所有偶数的和为：SUM=2550



## 实施步骤

在“命令”对话框中输入以下命令并按 Enter 键执行，新建程序文件。

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\ chengxu4_6
```

在程序编辑窗口输入以下代码：

```

USE D:\ 学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\ 学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
DO WHILE NOT EOF()
 xh=学号
 SELECT SelCourse
 GO TOP
LOCATE FOR 学号=xh
 cj= " "
 IF Found()
 DO CASE
 CASE 总成绩>=90 AND 总成绩<=100
 cj=" 优 "
 CASE 总成绩>=80 AND 总成绩<90
 cj=" 良 "
 CASE 总成绩>=70 AND 总成绩<80
 cj=" 中 "
 CASE 总成绩>=60 AND 总成绩<70
 cj=" 及格 "
 CASE 总成绩< 60
 cj=" 不及格 "
 ENDCASE
 ENDIF
 SELECT Student
 @10,10 SAY " 学号： "+学号
 @12,10 SAY " 姓名： "+姓名
 @14,10 SAY " 性别： "+性别
 @16,10 SAY " 专业： "+专业
 @18,10 SAY " 成绩： "+cj
WAIT
CLEAR
SKIP
ENDDO
CLOSE TABLES ALL

```

按 Ctrl+W 组合键保存程序文件。

运行程序文件 chengxu4\_6，在“命令”对话框中输入以下命令：

```
DO D:\学生成绩管理信息系统\APP\ chengxu4_6
```

程序运行结果如图 4-19 所示。



## 问题探究

1. 在应用 DO WHILE...ENDDO 循环结构时, 如何避免出现死循环?

DO WHILE...ENDDO 循环结构的循环体中, 一定要包含修改循环条件的、使循环能够结束的语句。chengxu4\_6 中的 SKIP 命令的作用是向表尾移动记录指针, 以便经过若干次循环后, 记录指针能够指向表尾, 使循环执行条件 NOT EOF() 为假, 达到结束循环的目的。如果没有这条命令, 运行程序时将进入无限循环 (即死循环), 程序不停地显示第一条记录。出现这种情况时, 可以按 Esc 键中断程序运行, 弹出程序错误提示框, 单击 “取消” 按钮, 终止程序运行。

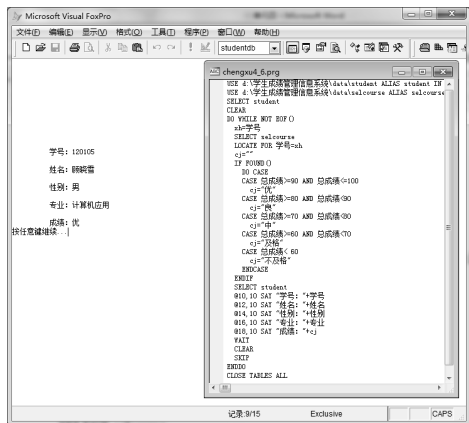


图 4-19 循环结构程序的运行结果

## 2. 如何使用记录循环语句 SCAN...ENDSCAN 改写 chengxu4\_6?

```
USE D:\ 学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\ 学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
SCAN
 xh=学号
 SELECT SelCourse
 GO TOP
 LOCATE FOR 学号=xh
 cj= " "
 IF Found()
 DO CASE
 CASE 总成绩>=90 AND 总成绩<=100
 cj= " 优 "
 CASE 总成绩>=80 AND 总成绩<90
 cj= " 良 "
 CASE 总成绩>=70 AND 总成绩<80
 cj= " 中 "
 CASE 总成绩>=60 AND 总成绩<70
 cj= " 及格 "
 CASE 总成绩< 60
 cj= " 不及格 "
 ENDCASE
 ENDIF
 SELECT Student
 @10,10 SAY " 学号: " +学号
 @12,10 SAY " 姓名: " +姓名
 @14,10 SAY " 性别: " +性别
 @16,10 SAY " 专业: " +专业
 @18,10 SAY " 成绩: " +cj
 WAIT
 CLEAR
ENDSCAN
CLOSE TABLES ALL
```



## 阅读材料

### 1. 结构化程序设计

结构化程序设计方法提出了一种解决复杂问题的程序设计思想：模块化、自顶向下、逐步求精。按照结构化程序设计方法，程序设计时由最上层的模块开始，一层一层地往下确定所需的模块，把大型程序逐层分解为一个一个功能相对独立的小模块，每个模块完成一个相对简单的功能，再把实现小模块功能的方法一步一步精确化，最终编写出程序。例如，可以把学籍管理系统分解为管理学生档案数据、管理学生成绩数据和管理借书数据 3 个大模块，再把管理档案数据模块细分成编辑数据、查询数据、统计数据、打印报表等小模块，并分别对各个模块设计下一层的小模块。例如，编辑数据模块又可以进一步细分为删除一个学生的记录和删除一批学生的记录等小模块，先编写和调试各个小模块的程序，再把它们组织成一个大程序，即可实现学籍管理的全部功能。

使用结构化程序设计方法编写程序时，首先仔细分析需要解决的问题，找出解决问题的方法和操作步骤，其次是用一组 Visual FoxPro 的命令实现操作步骤指定的功能，并将这些命令按照一定的结构组织在程序文件中即可。

### 2. 常用环境设置命令

#### (1) 关闭对话命令 (SET TALK ON/OFF)

命令格式：

```
SET TALK ON/OFF
```

功能：控制非输出性的执行结果是否在屏幕上显示，系统默认为 ON。

#### (2) 设置系统默认路径命令 (SET DEFAULT TO)

命令格式：

```
SET DEFAULT TO 默认路径
```

功能：用于设置系统默认打开文件的路径。

#### (3) 设置精确比较命令 (SET EXACT ON/OFF)

命令格式：

```
SET EXACT ON/OFF
```

功能：在进行字符比较时，是否需要精确比较。

说明：ON 表示需要精确比较，OFF 表示不需要。系统默认值为 OFF。

#### (4) 设置保护状态命令 (SET SAFETY ON/OFF)

命令格式：

```
SET SAFETY ON/OFF
```

功能：屏蔽或处理有删除标记的记录。

说明：当命令格式中选择 ON 时，系统将不再对有删除标记的记录进行操作（建立索引的命令除外），选择 OFF 时，对这些记录仍然可以进行显示等操作，系统默认为 OFF。

#### (5) 设置删除标记的有效性命令 (SET DELETED ON/OFF)

命令格式：

```
SET DELETED ON/OFF
```

说明：

如果设置为 ON 状态，不对带有删除标记的记录进行操作；设为 OFF，则对带有删除标记



的记录也进行操作。注意：对于当前记录或由 RECORD N 特指的记录，不管是否有删除标记，不管 DELETED 置于何种状态，一律可进行操作。此外，索引也忽略记录的删除标记。

#### (6) 设置 ESC 键的有效性命令 (SET ESCAPE OFF)

命令格式：

```
SET ESCAPE OFF
```

说明：

该命令决定是否通过按 Esc 键中断程序和命令的运行。

### 3. 程序调试基础

初学程序设计时，编写的程序常常会出现各种错误。因此，每个编程者都需要掌握一定的程序调试方法，以便能快速、准确地发现错误、改正错误。程序中的错误通常分为两类：语法错误和逻辑错误。程序中命令格式、命令的关键字或程序结构出现错误将导致语法错误，命令或表达式使用不当将导致逻辑错误。

程序调试就是发现并改正程序中的语法错误和逻辑错误。

运行程序时，Visual FoxPro 可以自动检查语法错误，并根据错误类型给出相应的出错信息。但 Visual FoxPro 不能检查程序的逻辑错误，逻辑错误需要编程者自己检查。

Visual FoxPro 提供了完善的程序调试功能，用户可以使用程序调试器单步跟踪程序运行，通过观察程序运行过程找出程序的错误，也可以使用调试程序的命令设置断点、输出中间结果等，还可以记录和分析程序代码的执行情况。

选择“程序”“调试器”命令，可以打开调试器窗口。单击“调试器”工具栏中的“打开”按钮或选择“测试”“程序”命令，可以打开需要调试的程序。按 F8 键或选择“调试”→“单步跟踪”命令，可以单步跟踪运行程序。通过单步跟踪程序运行，观察程序中变量和表达式值的变化，可以发现程序的逻辑错误。其他调试程序的方法可以参看帮助信息。

### 4. 程序设计中常用的辅助命令

#### (1) 清除内存变量命令 (CLEAR ALL)

命令格式：

```
CLEAR ALL
```

功能：关闭打开的数据库、表及索引文件，清除用户自定义的内存变量和对象。

#### (2) 返回命令 (RETURN)

命令格式：

```
RETURN
```

功能：结束本程序的运行。

说明：如果当前程序无上级程序，该命令用于结束程序的运行，返回“命令”对话框；如果当前程序是一个子程序，该命令用于结束程序的运行，返回到调用该程序的上级程序中。

#### (3) 终止程序执行命令 (CANCEL)

命令格式：

```
CANCEL
```

功能：终止程序执行并关闭所有打开的文件，返回“命令”对话框。

#### (4) 退出系统命令 (QUIT)

命令格式：

```
QUIT
```

功能：终止程序执行并关闭所有打开的文件，退出 Visual FoxPro 系统。

### 5. 针对集合的循环

针对集合的循环用于对集合中的每个元素进行相同的操作，是 FOR 循环结构的一种，其格式如下：

```
FOR EACH <变量> IN <集合变量>
<语句序列>
[LOOP]
[EXIT]
ENDFOR/NEXT
```

功能：用<变量>依次引用<集合变量>中的每个元素，执行循环体中的语句，直到<集合变量>中的所有元素都遍历完，或者遇到 EXIT 命令，结束循环。

说明：

- <集合变量>可以是一个 Visual FoxPro 数组、OLE 数组、一个 Visual FoxPro 集合或 OLE 集合。

- LOOP 与 EXIT 语句的作用与在 DO WHILE 语句中的作用相同。

集合循环语句的应用如下：显示 1~100 以内的偶数。

```
DECLARE MyArray(100)
FOR I=1 TO 100
 MyArray(I)=I
ENDFOR
FOR EACH I IN MyArray
 IF MOD(I,2)==0 THEN
 ?I
 ENDIF
ENDFOR
```



## 项目小结

程序文件的扩展名为“.prg”，程序文件在编译后生成“.fxp”文件，对某一程序文件进行修改后，则生成一个扩展名为“.bak”的文件，用来保存修改前的程序文件内容。

本项目介绍了面向过程的程序设计方法，在 Visual FoxPro 中创建、编辑、运行程序文件的操作方法；交互式输入命令和格式化输出命令的应用；顺序结构、选择结构、循环结构 3 种基本的程序设计结构及相关命令语句。

- 顺序结构是指程序按照输入的代码顺序执行，没有分支和转向。

- 选择结构是指根据一个或多个条件，确定程序走向，使用的语句有 IF...ELSE...ENDIF、DO CASE...ENDCASE 语句。

- 循环结构解决需反复执行的某程序段的编程问题。使用的语句有 DO WHILE...ENDDO、SCAN...ENDSCAN、FOR...ENDFOR/NEXT。

在实际的程序设计中，往往需要综合应用这 3 种结构。



## 上机实习

1. 设计程序，显示“这是我的第一个 Visual FoxPro 程序……”。



2. 编写程序, 请用户输入学生姓名, 查询并输出该学生的学号、性别、专业。
3. 编写程序, 请用户输入圆的半径值, 计算输出圆的周长和面积。
4. 设计一个程序, 用于判断用户输入的数值是奇数还是偶数。
5. 编写程序, 任意输入 3 个数, 输出其中最大的数。
6. 编写程序, 任意输入 3 个数, 按由小到大的顺序输出。
7. 输入 10 个整数, 分别统计正数、负数和零的个数。
8. 编写程序, 计算下列分段函数:

$$f(x)=\begin{cases} x+2 & (x \leq 0) \\ x^2+3 & (0 < x < 5) \\ 2x-6 & (5 \leq x \leq 15) \\ x^2-x & (15 < x) \end{cases}$$

9. 编写一个程序, 根据当前系统日期显示当前月份的天数。
10. 编写程序, 判断某一年是否为闰年 (闰年的年号能被 4 整除但不能被 100 整除, 或者能被 400 整除)。
11. 编写程序实现以下转换。  
输入: 0~6; 对应转换为: 星期日、星期一、.....、星期六  
(要求若输入的数据在 0~6 范围之外, 显示“输入有误!”)
12. 编写程序, 根据不同时间, 显示问候语:  
5:00~8:00, 显示“早上好”  
8:00~12:00, 显示“上午好”  
12:00~18:00, 显示“下午好”  
其他时间, 显示“晚上好”
13. 依次显示 1~10 的自然数。
14. 计算自然数的和, 当和大于 500 时停止。
15. 设计程序, 统计表 Student 中女生的人数。
16. 设计程序, 逐条显示表 Student 中籍贯为山东的学生记录。
17. 设计程序, 输入年份, 显示在该年份出生的学生记录。
18. 编写程序, 求  $S=12+22+32+\dots+1002$ 。
19. 找出 3~100 之间所有的素数。
20. 设计程序, 输入  $N$ , 计算  $N!(1 \times 2 \times 3 \times \dots \times N)$ 。



## 知识巩固

### 一、选择题

1. 下列方法不能运行程序的是 ( )。
  - A. 在项目管理器中选择某个程序文件选项和单击“运行”按钮
  - B. 使用“打开”对话框打开程序文件
  - C. 选择“程序” “运行”命令
  - D. 执行运行程序的 DO 命令
2. 若程序出现语法错误, 单击“程序错误”对话框中的 ( ) 按钮可以终止程序运行。

- A. 终止                      B. 取消                      C. 挂起                      D. 忽略
3. 下列叙述正确的是 (      )。
- A. WAIT 命令只能接收一个字符              B. ACCEPT 命令只能接受一个字符  
C. INPUT 命令不能接收字符型数据              D. INPUT 命令只能接受一个字符
4. 命令文件的扩展名是 (      )。
- A. .idx                      B. .frm                      C. .prg                      D. .dbf
5. 有关 SCAN 循环结构, 叙述正确的是 (      )。
- A. SCAN 循环中的 LOOP 语句, 可将程序流程指向循环开始语句 SCAN, 首先判断 EOF()函数的真假  
B. 在使用 SCAN 循环结构时, 必须打开某一个数据库  
C. SCAN 循环的循环体中必须写有 SKIP 语句  
D. SCAN 循环结构, 如果省略了范围子句和 FOR 条件子句, 则直接退出循环
6. 在 FOR...ENDFOR 循环结构中, 如果省略步长, 则系统默认步长为 (      )。
- A. 0                      B. -1                      C. 1                      D. 2
7. 在 DO WHILE...ENDDO 循环结构中, LOOP 命令的作用是 (      ), EXIT 命令的作用是 (      )。
- A. 退出程序, 返回程序开始处  
B. 转移到 DO WHILE 语句行, 开始下一个循环和判断  
C. 终止循环, 将控制转移到本循环结构 ENDDO 后面的第一条语句继续执行  
D. 终止程序执行
8. 下列命令中, 不能使程序退出循环的是 (      )。
- A. LOOP                      B. EXIT                      C. QUIT                      D. RETURN
9. 在 Visual FoxPro 中 DO CASE...ENDCASE 属于 (      ) 结构。
- A. 顺序                      B. 选择                      C. 循环                      D. 模块
10. 在 Visual FoxPro 中的 3 种循环结构中, 当循环次数为已知时, 应选用的语句是 (      )。
- A. DO WHILE              B. SCAN                      C. FOR                      D. LOOP

## 二、填空题

1. 结构化程序设计语言的基本程序结构是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 使用\_\_\_\_\_命令创建程序文件。
3. 为程序中的语句行添加注释可以使用\_\_\_\_\_符号或\_\_\_\_\_命令, 在语句行后面添加注释可以使用\_\_\_\_\_符号。
4. 清除屏幕显示信息的命令是\_\_\_\_\_。
5. 清除用户自定义内存变量和对象的命令是\_\_\_\_\_。
6. 选择\_\_\_\_\_命令, 可以打开调试器窗口调试程序。

## 三、简答题

1. 简述 MODIFY COMMAND 命令和 DO 命令的功能和使用时的注意事项。
2. 结构化程序设计的 3 种基本结构是什么?
3. EXIT 和 LOOP 命令的作用分别是什么?



#### 4. 简述顺序结构程序设计的一般步骤。

#### 四、写出下面程序的运行结果

##### 1. 程序如下：

```
SET TALK OFF
CLEAR
FOR i=1 TO 5
? Replicate('*',2*i-1)

ENDFOR
RETURN
```

&&返回将字符串“\*”重复  $2*i-1$  次后的字符串

##### 2. 程序如下：

```
SET TALK OFF
CLEAR
B=6
A='7'
?A,B
```

##### 3. 程序如下：

```
SET TALK OFF
CLEAR
STORE 0 TO B
FOR A=0 TO 10
IF MOD(A,2)==0
B=B+1
ENDIF
ENDFOR
?A,B
SET TALK ON
RETURN
```

##### 4. 程序如下：

```
SET TALK OFF
CLEAR
I=1
X=0
DO WHILE I<=4
X=X+I
I=I+1
ENDDO
?X
RETURN
```

##### 5. 程序如下：

```
SET TALK OFF
CLEAR
STORE 0 TO X,Y,S1,S2,S3
DO WHILE X<10
X=X+1
DO CASE
CASE Int(X/2)=X/2
S1=S1+X/2
CASE Mod(X,3)=0
```



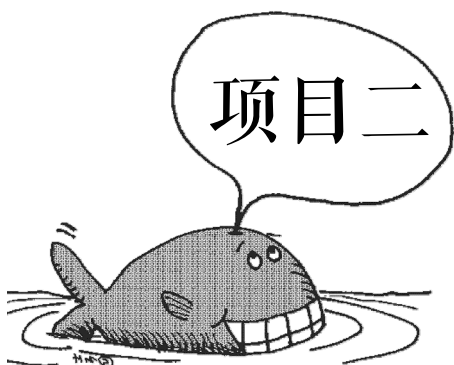
```
S2=S2+X/3
CASE Int (X/2) <>X/2
S3=S3+1
ENDCASE
ENDDO
?S1,S2,S3
SET TALK ON
RETURN
```

#### 6. 程序如下：

```
SET TALK OFF
CLEAR
STORE 1 TO X
STORE 20 TO Y
DO WHILE X<=Y
IF Int (X/2) <>X/2
X=1+X^2
Y=Y+1
LOOP
ELSE
X=X+1
ENDIF
ENDDO
?X
?Y
SET TALK ON
RETURN
```

#### 7. 程序如下：

```
SET TALK OFF
CLEAR
S=0
FOR I=0 TO 40 STEP 4
S=S+I
ENDFOR
?S
SET TALK ON
RETURN
```



# 模块化编写“输出指定学号的学生信息”程序

## 项目说明

在 Visual FoxPro 中，过程和自定义函数可以把经常使用的功能代码独立出来。当在程序中需要使用相同功能的代码序列时，可以反复调用已经编写好的代码，从而有利于程序的维护，对程序的修改只需修改过程或函数，主程序无须修改。

本项目重点介绍模块化程序设计中常用的子程序、过程和自定义函数的结构和调用方法。此外，还应该了解在主程序和子程序之间参数的传递过程和变量的作用域。综合这些知识，才能够设计出高效、实用的数据库应用程序。

本项目包含 3 个任务：

任务一：编写“格式化显示学生信息”子程序，并在主程序中调用。

任务二：将子程序改写为过程文件，并在主程序中调用。

任务三：将获取学生成绩的过程文件“GetCj”转换成函数文件。

## 任务一 编写“格式化显示学生信息”子程序，并在主程序中调用

## 任务目标

理解子程序与主程序的概念。

能够熟练创建、编辑子程序及返回主程序。

能够在主程序中调用子程序。

任务描述

编写子程序 SubPrg.prg，实现“格式化输出某个学生信息和成绩”的功能；编写主程序 main.prg，通过调用 SubPrg.prg，格式化输出 Student.dbf 表中所有学生的信息和成绩。

相关知识

在应用程序开发过程中，一个应用程序一般由多个模块组成，每个模块完成特定的功能，并且相对独立，同时可以为其他模块调用，也可以调用其他模块。通常把这种具有相对独立和通用的程序模块称为子程序，调用其他模块的程序模块称为主程序。

1. 创建子程序

子程序和其他 Visual FoxPro 程序一样，可以使用 MODIFY COMMAND 命令来创建、修改或保存，也可以使用其他编辑软件来创建和修改。子程序以独立的程序文件方式存放在磁盘上，每个子程序的默认扩展名为“.prg”。子程序一般以 RETURN 命令结束，才能保证返回到调用它的主程序，其他的返回语句如下。

|                   |                   |
|-------------------|-------------------|
| RETURN TO MASTER。 | 返回到最高一级的主程序。      |
| RETRY。            | 返回到调用语句。          |
| CANCEL：           | 结束本程序。            |
| QUIT：             | 退出 Visual FoxPro。 |

2. 调用子程序

调用子程序可以使用 DO 命令，其格式如下：

```
DO <子程序名> [WITH<参数清单>]
```

功能：调用执行<子程序名>所指定的程序文件。

说明：

- <参数清单>中的参数可以是常量，也可以是表达式，但如果为变量，则必须具有初值。
- 子程序还可以调用其他子程序，构成子程序调用的嵌套。

如图 4-20 所示为 3 个程序之间的调用和返回点。

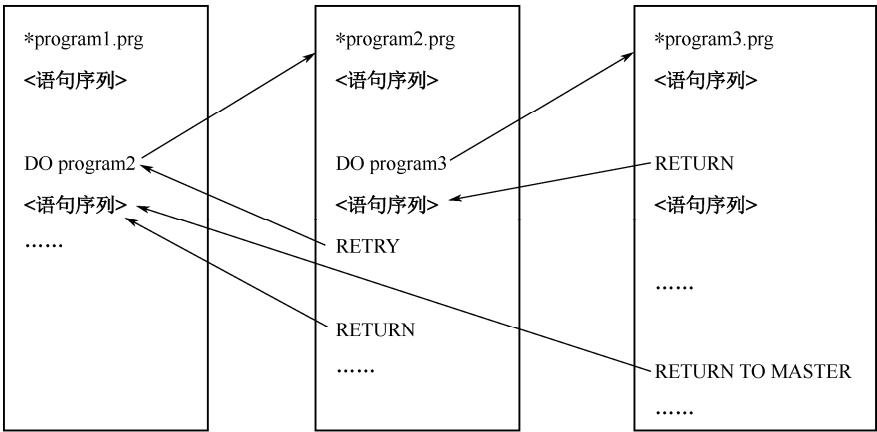


图 4-20 程序间的调用与返回点



## 实施步骤

新建子程序文件 SubPrg.prg，在“命令”对话框中输入以下命令：

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\子程序\SubPrg
```

子程序完成“显示一名学生的自然信息和成绩信息”的功能。在子程序文件 SubPrg.prg 中输入如下代码：

```
SELECT Student
xh=学号
cj=" "
SELECT SelCourse
GO TOP
LOCATE FOR 学号=xh
IF Found()
 DO CASE
 CASE 总成绩>=90 AND 总成绩<=100
 cj=" 优 "
 CASE 总成绩>=80 AND 总成绩<90
 cj=" 良 "
 CASE 总成绩>=70 AND 总成绩<80
 cj=" 中 "
 CASE 总成绩>=60 AND 总成绩<70
 cj=" 及格 "
 CASE 总成绩< 60
 cj=" 不及格 "
 ENDCASE
ENDIF
SELECT Student
@10,10 SAY " 学号： " +学号
@12,10 SAY " 姓名： " +姓名
@14,10 SAY " 性别： " +性别
@16,10 SAY " 专业： " +专业
@18,10 SAY " 成绩： " +cj
```

保存子程序文件 SubPrg.prg。

新建主程序文件 main.prg，在“命令”对话框中输入如下命令：

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\子程序\main
```

主程序通过对所有学生记录进行遍历，并分别调用子程序，完成“显示所有学生自然信息和成绩信息”的功能。在主程序文件 main.prg 中输入如下代码：

```
USE D:\学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
DO WHILE NOT EOF()
DO D:\学生成绩管理信息系统\APP\子程序\SubPrg
WAIT
 CLEAR
 SKIP
ENDDO
CLOSE TABLES ALL
```

保存并运行主程序文件 main.prg，检验调用子程序是否成功。

## 任务二 将子程序改写为过程文件，并在主程序中调用

### 任务目标

理解过程、过程文件的概念。  
能够熟练定义和调用过程。

### 任务描述

创建过程文件 DispOneStu.prg，编写过程 DispStu，格式化输出某个学生的信息和成绩；编写过程 GetCj，得到某个学生的成绩。设计主程序文件 DispAllStu.prg，分别调用过程 DispStu、GetCj，格式化输出表 Student.dbf 中所有学生的自然信息和成绩信息。

### 相关知识

将多个子程序以“过程”的形式编写在同一个文件中，这样的文件称为过程文件。由于过程文件是由多个子程序组成的一个较大的文件，只要对该文件读取一次，就能把若干个子程序都读进来，从而避免了同时打开多个文件和频繁读写磁盘文件的操作，提高了系统运行效率。

#### 1. 创建过程

创建过程文件和创建其他程序文件一样，可以使用 MODIFY COMMAND<过程文件名>来实现，默认的扩展名为“.prg”。过程文件的结构如下：

```
PROCEDURE 过程名 1 [PARAMETER 参数表]
 命令序列 1
[RETURN [<表达式>]]
ENDPROC
PROCEDURE 过程名 2 [PARAMETER 参数表]
 命令序列 2
[RETURN [<表达式>]]
ENDPROC
.....
```

说明：

- 过程文件中的每一个过程都必须以 PROCEDURE 语句开头，后面跟过程名，过程名没有扩展名。
- ENDPROC 表示一个过程的结束。如果省略 ENDPROC 语句，则过程结束于下一条 PROCEDURE 命令或文件结尾处。
- 当过程执行到 RETURN 语句时，控制将转到调用程序（或“命令”对话框），并返回表达式的值。如果省略 RETURN 命令，则在过程结束处自动执行一条隐含的 RETURN 命令。

#### 2. 调用过程

过程文件里只包含过程，这些过程可以被其他程序调用。但在调用过程文件中的过程之前首先要打开过程文件，打开过程文件的命令格式如下：



SET PROCEDURE TO [ 过程文件名 1 , [ 过程文件名 2 , ... ] ][ADDITIVE]

功能：关闭或打开过程文件。

说明：

- 该命令打开一个或多个过程文件。
- 选择 ADDITIVE 选项，在打开过程文件时，不关闭先前打开的过程文件。
- 该命令一般在主程序中使用，或至少放在调用过程的前面。过程文件一旦被打开，过程文件中的所有子过程也被打开，也就是说被调入内存。对于一个打开的过程文件，可随时被调用其中的任何子过程，从而减少了访问磁盘的次数，提高了程序运行的速度。

Visual FoxPro 系统中，过程可以单独存放在过程文件中，也可以存放在程序文件的末尾，与程序一起保存在程序文件中。存放在程序后面的过程，可以直接使用调用子程序的 DO 命令调用。要调用存放在过程文件中的过程，需要首先打开过程文件，再使用 DO 命令调用其中的过程。

当过程文件中的过程调用结束后，应及时关闭过程文件。关闭过程文件的命令格式有以下两种。

第一种命令格式如下：

```
SET PROCEDURE TO
```

功能：关闭打开的所有过程文件。

第二种命令如下：

```
RELEASE PROCEDURE 过程文件名 1 [, 过程文件名 2 , ...]
```

功能：关闭指定的过程文件。



## 实施步骤

新建过程文件 DispOneStu.prg，在“命令”对话框中输入如下代码：

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\过程\DispOneStu
```

在过程文件 DispOneStu.prg 中编写过程 DispStu、GetCj，其中 DispStu 过程功能是显示当前记录学生的自然信息和成绩信息；GetCj 过程功能是获得学号为 xh 的学生成绩信息。代码如下：

```
PROCEDURE DispStu
PARAMETER xh, xm, xb, zy, cj
 @10,10 SAY "学号：" + xh
 @12,10 SAY "姓名：" + xm
 @14,10 SAY "性别：" + xb
 @16,10 SAY "专业：" + zy
 @18,10 SAY "成绩：" + cj
 RETURN
ENDPROC
PROCEDURE GetCj
PARAMETER xh, cj
 cj = " "
 SELECT SelCourse
 GO TOP
 LOCATE FOR 学号=xh
 IF Found()
 DO CASE
CASE 总成绩>=90 AND 总成绩<=100
```

```

 cj= " 优 "
CASE 总成绩>=80 AND 总成绩<90
 cj= " 良 "
CASE 总成绩>=70 AND 总成绩<80
 cj= " 中 "
CASE 总成绩>=60 AND 总成绩<70
 cj= " 及格 "
CASE 总成绩< 60
 cj= " 不及格 "
ENDCASE
ENDIF
RETURN
ENDPROC

```

保存过程文件 DispOneStu.prg。

新建主程序文件 DispAllStu.prg，主程序功能是显示所有学生的自然信息和成绩信息。

在“命令”对话框中书写如下命令：

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\过程\DispAllStu
```

在程序文件 DispAllStu.prg 中输入以下语句，调用过程文件 DispOneStu.prg 中的过程。

```

SET PROCEDURE TO D:\学生成绩管理信息系统\APP\函数\DispOneStu.prg
USE D:\学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
DO WHILE NOT EOF() &&逐一记录显示所有学生的自然信息和成绩信息
 cj= " "
 DO GetCj WITH 学号,cj &&调用子程序，将成绩通过变量 cj 返回
SELECT Student
 DO DispStu WITH 学号,姓名,性别,专业,cj &&子程序显示当前记录学生的自然信息和成绩信息

 WAIT
 CLEAR
 SKIP
ENDDO
CLOSE TABLES ALL
SET PROCEDURE TO

```

保存并运行主程序文件 DispAllStu.prg，验证程序结果。

## 任务三 将获取学生成绩的过程文件“GetCj”转换成函数文件

### 任务目标

理解用户自定义函数的概念。

能够熟练创建和调用自定义函数。

了解参数传递的过程。

掌握多模块程序设计中变量的作用域。



## 任务描述

创建过程文件 DispOneStu.prg, 编写过程 DispStu, 格式化输出某个学生的自然信息和成绩信息 编写函数 GetCj, 返回某个学生的成绩。设计主程序文件 DispAllStu.prg, 调用过程 DispStu、函数 GetCj, 格式化输出表 Student.dbf 中所有学生的自然信息和成绩信息。

## 相关知识

Visual FoxPro 系统提供了众多的系统函数, 为处理数据和编写程序提供了很大的方便。此外, Visual FoxPro 系统还允许用户将频繁使用的程序段定义成为独立的函数, 成为用户自定义函数, 以便减少程序的数量和复杂性, 提高程序的可读性和重用性。

### 1. 自定义函数的创建

用户自定义函数的常用格式如下:

```
FUNCTION <函数名>
[PARAMETER <变量名列表>]
[<语句序列>]
[<RETURN> <表达式>]
ENDFUNC
```

说明:

- PARAMETER 后的变量名列表称为形式参数, 变量名之间由英文逗号分开。
- RETURN 语句用于返回函数值, 其中 表达式 的值就是函数的值; 如果省略该语句, 函数的返回值为真 (.T.)。
- 自定义函数可以包含在调用程序中, 也可作为独立文件保存。

### 2. 自定义函数的调用

可以在任何与函数返回值类型兼容的表达式中调用函数。自定义函数的调用与 Visual FoxPro 系统预定义函数的调用格式相同, 格式如下:

```
<函数名> (<参数表达式列表>)
```

说明:

- <参数表达式列表>称为实际参数, 实参中的各表达式之间由英文逗号分开, 实参的个数必须与调用函数中形参的个数相同, 且一一对应。
- 调用过程文件中的函数, 要先用 SET PROCEDURE TO 过程文件名 命令打开过程文件, 调用自定义函数完成后, 再用 SET PROCEDURE TO 命令关闭过程文件。



## 实施步骤

新建过程文件 DispOneStu.prg, 在“命令”对话框中输入以下代码:

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\函数\DispOneStu
```

在程序文件 DispOneStu.prg 中编写过程 DispStu 和函数 GetCj, 代码如下:

```
PROCEDURE DispStu
PARAMETER xh, xm, xb, zy, cj
 @10,10 SAY "学号: " + xh
 @12,10 SAY "姓名: " + xm
```



```

@14,10 SAY "性别：" +xb
@16,10 SAY "专业：" +zy
@18,10 SAY "成绩：" +cj
RETURN
ENDPROC
FUNCTION GetCj &&将 GetCj 定义为函数
PARAMETER xh
 cj= " "
 SELECT SelCourse
 GO TOP
 LOCATE FOR 学号=xh
 IF Found()
 DO CASE
 CASE 总成绩>=90 AND 总成绩<=100
 cj=" 优 "
 CASE 总成绩>=80 AND 总成绩<90
 cj=" 良 "
 CASE 总成绩>=70 AND 总成绩<80
 cj=" 中 "
 CASE 总成绩>=60 AND 总成绩<70
 cj=" 及格 "
 CASE 总成绩< 60
 cj=" 不及格 "
 ENDCASE
 ENDIF
 RETURN cj
ENDFUNC

```

保存过程文件 DispOneStu.prg。

新建主程序文件 DispAllStu.prg，在“命令”对话框中书写以下命令：

```
MODIFY COMMAND D:\学生成绩管理信息系统\APP\函数\DispAllStu
```

在程序文件 DispAllStu.prg 中输入以下语句，调用过程文件 DispOneStu.prg 中的过程和函数。

```

SET PROCEDURE TO D:\学生成绩管理信息系统\APP\过程\DispOneStu.prg
USE D:\学生成绩管理信息系统\DATA\SelCourse ALIAS SelCourse
USE D:\学生成绩管理信息系统\DATA\Student ALIAS Student
CLEAR
DO WHILE NOT EOF()
 cj= " "
 cj=GetCj(学号) &&调用函数，返回值赋给变量 cj
 SELECT Student
 DO DispStu WITH 学号,姓名,性别,专业,cj
 WAIT
 CLEAR
 SKIP
ENDDO
CLOSE TABLES ALL
SET PROCEDURE TO

```

保存并运行主程序文件 DispAllStu.prg，格式化输出表 Student.dbf 中所有学生的自然信息和成绩信息。



## 问题探究

子程序、过程与自定义函数有何区别？

- 子程序是独立的程序文件；过程和自定义函数既可以放在独立的程序文件中，也可以放在主程序文件中。

- 一个子程序文件完成独立的功能；一个过程文件可以存放多个过程或自定义函数。

- 子程序和过程的调用使用 DO <子程序 / 过程名> [WITH<参数表>]。自定义函数的调用方法和系统预定义函数相同，采用<函数名>(<参数表达式列表>)的方式。过程和自定义函数如果存放在程序文件中，调用前还需使用 SET PROCEDURE TO [ 过程文件名 1 ] 打开程序文件。

- 子程序、过程没有返回值，主要目的是完成指定的操作；自定义函数可以有返回值，格式为 RETURN[<返回值 (常量、变量或表达式)>]，主要目的是获得一个计算结果值，这个结果值最终通过函数名返回给调用者。



## 阅读材料

### 1. 参数传递

主程序调用过程（子程序）或函数时，常常需要进行参数传递，把调用程序中的数据传递给过程（子程序）或函数。Visual FoxPro 提供了程序间参数传递的功能，用于程序之间的数据交换。调用程序所传出的参数称为实际参数，被调用程序所接收的参数称为形式参数。

实现带参数程序的调用，主程序调用过程（子程序）的命令格式如下：

```
DO <子程序名> [WITH<参数表>]
```

当调用程序进行参数传递时，被调用程序中的第一个语句应为参数接收语句，其格式如下：

```
PARAMETER 参数表
```

说明：

- 该语句的功能是接收 DO...WITH<参数表>调用命令传递来的参数值。
- 如果调用的是过程文件，语句 PARAMETER 参数表 一般放在 PROCEDURE 语句之后；如果是自定义函数，则放在 FUNCTION 语句之后。参数调用时，WITH<参数表>中的参数与调用程序中的参数保持一致（数量相同，类型一致，但参数名可以不同）。

### 2. 变量的作用域

在多模块程序中，每一个内存变量都有自己的有效范围（生存期），也就是说该变量在多大范围内能够被识别，通常我们称之为作用域。

按照内存变量的作用域不同，可将其分为 3 类。

公共变量：PUBLIC VARIABLE。

私有变量：PRIVATE VARIABLE。

本地（局部）变量：LOCAL VARIABLE。

#### (1) 公共变量

公共变量（又称全局变量）是指在上下各级程序中都可以使用的内存变量。公共变量必须使用 PUBLIC 命令来定义，其格式如下：

```
PUBLIC 内存变量表
```

功能：将 内存变量表 中的变量设置为公共变量，并将这些变量的初值均赋为 “.F.”。

说明：

- 在“命令”对话框中定义的内存变量（包括数组）和在程序中用 PUBLIC 命令定义的内存变量都是公共变量。无论哪一级程序中，使用 PUBLIC 定义的变量，均可成为公共变量。

- 即使程序运行结束，这些变量也不会自动清除，要依靠 CLEAR、RELEASE 等语句来清除。

## （2）私有变量

程序中如果没有对内存变量的属性加以说明，则这些变量就是私有变量。私有变量只能在本级程序及以下各级程序（子程序）中使用。当调用程序中的私有变量在过程（子程序）中改变了其值，过程（子程序）运行结束返回调用程序时，其值也随之变化。但在过程（子程序）中的私有变量，当过程（子程序）运行结束后，该内存变量立即被清除，不能返回上级调用程序。

私有变量的声明格式如下：

PRIVATE 内存变量表

功能：声明私有变量并隐藏上级模块中的同名变量，直到该变量的程序、过程或自定义函数执行结束后，才恢复使用先前隐藏的同名变量。

说明：

- 私有变量的作用范围为定义它的模块及其下层模块，运行结束后自动清除。

当一个大的应用系统由多人进行开发时，很可能因为变量名相同而造成程序的错误。如果个人将自己所使用的变量声明为 PRIVATE，就可以有效地避免发生混淆。

## （3）本地（局部）变量

局部变量只能在建立它的程序中使用，不能在上级或下级程序中使用。当它所处的程序运行结束时，局部变量立即被清除。局部变量使用 LOCAL 命令来定义，其格式如下：

LOCAL 内存变量表

说明：

- 该命令定义的内存变量为局部变量。局部变量要先定义后使用。

- 使用时自动隐藏上级变量，模块结束后自动清除，进入下级模块后自动隐藏。



## 项目小结

在应用程序开发过程中，Visual FoxPro 的一个应用程序一般由多个模块组成，每个模块完成特定的功能并相对独立，同时可以被其他模块调用，也可以调用其他模块。本单元介绍了子程序、过程和自定义函数 3 种模块的新建、编辑与调用过程。正确应用多模块程序设计，可以提高程序设计的有效性和程序的运行速度。



## 上机实习

1. 编写一个计算  $n!$  的函数，在主程序中调用该函数，计算  $10!$ 。
2. 编写一个求和  $1+2+\dots+n$  的过程，并调用此过程求  $S=1+(1+2)+(1+2+3)+\dots+(1+2+\dots+20)$  的值。
3. 编写求绝对值函数，并在主程序中调用，请用户输入整数，输出该整数的绝对值。
4. 利用子程序和参数传递计算  $10!+15!+20!+25!$ 。
5. 编写过程判断  $n$  是否为素数，然后调用该过程求  $2\sim 1000$  内的素数。



## 知识巩固

### 一、选择题

1. 下列说法中正确的是 ( )。  
A. 若函数不带参数, 则调用时函数名后面的圆括号可以省略  
B. 函数若有多个参数, 则各参数间应用空格隔开  
C. 调用函数时, 参数的类型、个数和顺序不一定要一致  
D. 调用函数时, 函数名后的圆括号不论有无参数都不能省略
2. 命令 SET DELETE OFF 的作用是 ( )。  
A. 禁止删除数据表中的记录  
B. 使删除标记对任何命令无影响  
C. 取消删除记录  
D. 执行删除命令时, 系统不提示任何信息
3. 用户自定义函数或过程中接收参数, 应使用 ( ) 命令。  
A. PROCEDURE  
B. FUNCTION  
C. WITH  
D. PARAMETERS
4. 在程序中调用另一程序, 使用命令 ( )。  
A. CALL 被调用程序  
B. LOAD 被调用程序  
C. DO 被调用程序  
D. PROCEDURE 被调用程序
5. 调用程序和被调用程序之间的参数传递要求 ( )。  
A. 参数必须为字符型  
B. 调用程序的 WITH 所给的参数必须是变量  
C. 被调用程序中不能改变 PARAMETER  
D. 调用程序 WITH 所带的参数必须与被调用程序的参数一一对应

### 二、填空题

1. 自定义函数过程中接受参数, 应使用\_\_\_\_\_命令。
2. 一个过程必须以\_\_\_\_\_开头, 为了调用该过程, 必须通过\_\_\_\_\_打开该过程文件。
3. 自定义函数必须以\_\_\_\_\_开头。
4. 在命令文件中, 调用另一个文件的命令是\_\_\_\_\_。
5. 子程序中的 RETURN TO MASTER 命令功能是\_\_\_\_\_。
6. 在 Visual FoxPro 中, 可以使用 PUBLIC、\_\_\_\_\_和\_\_\_\_\_命令定义变量的作用范围。

### 三、写出下列程序的运行结果

1. 程序如下:

```
PUBLIC T
CLEAR
S=0
FOR K=1 TO 10
DO JC
S=S+T
```

```

ENDFOR
?"S=",S
RETURN
PROCEDURE JC
PRIVATE S
T=1
FOR S=1 TO K
T=T*S
ENDFOR
?Str(K,2)+"!" +Str(T,8)
RETURN

```

## 2. 程序如下：

```

SET TALK OFF
m=1
n=2
DO B
?"m=",m,"n=",n
RETURN
PROCEDURE B
PRIVATE m
m=3
n=4
RETURN

```

## 3. 程序如下：

```

SET TALK OFF
PUBLIC a1
x=1
y=2
a1=40
DO APROC
?a1,a2,x,y
RETURN
PROCEDURE APROC
PRIVATE x,a1
PUBLIC a2
x=10
y=20
a1=33
a2=22
?a1,a2,x,y
RETURN

```

## 4. 程序如下：

```

SET TALK OFF
PUBLIC I,J,K
I=1
DO PROC1
?"I="+STR(I,2)
J=1
K=1
DO PROC2
?"J="+STR(J,2)
?"K="+STR(K,2)

```



```
RETURN
PROCEDURE PROC1
I=I*2+1
RETURN
PROCEDURE PROC2
PRIVATE J
J=I*2+1
K=K*2+1
DO PROC3
RETURN
PROCEDURE PROC3
K=K*K
RETURN
```

**5. 主程序 Z.prg 如下：**

```
SET TALK OFF
STORE 1 TO X1,X2,X3
X1=X1+1
DO Z1
?X1+X2+X3
RETURN
SET TALK ON
```

**子程序 Z1.prg 如下：**

```
X2=X2+1
DO Z2
X1=X1+1
RETURN
```

**子程序 Z2.prg 如下：**

```
X3=X3+1
RETURN TO MASTER
PROCEDURE APROC
PRIVATE x,a1
PUBLIC a2
x=10
y=20
a1=33
a2=22
?a1,a2,x,y
RETURN
```



# 单元五

## 系统设计与实现

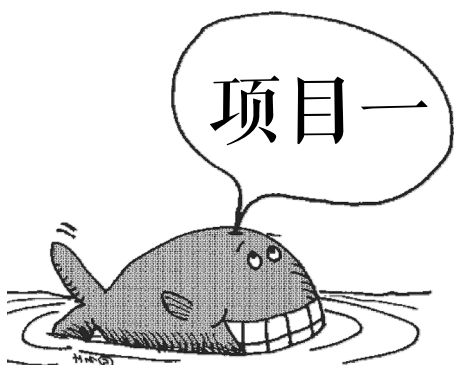
### 知识目标

- 1) 了解面向对象的基本概念。
- 2) 了解面向对象程序的基本思想。
- 3) 熟练掌握表单及常用控件的常用属性、事件和方法。常用控件包括标签、文本框、命令按钮、命令按钮组、选项按钮组、编辑框、列表框、组合框、计时器、表格等。
- 4) 掌握报表的设计与制作方法。
- 5) 掌握菜单的设计与制作方法。

### 技能目标

- 1) 能够制作管理信息系统的主表单。
- 2) 能够制作管理信息系统的登录界面。
- 3) 能够制作管理信息系统的主表单菜单。
- 4) 能够制作管理信息系统的各个模块。
- 5) 能够对管理信息系统进行连编和发布。





# 设计并制作“学生成绩管理信息系统”的主表单

## 项目说明

随着计算机技术的发展，人们对于数据的管理与处理已实现了自动化，各行各业对数据的管理都已经进入到了数据库系统阶段。同时随着学校扩招政策的深入实施，学生人数越来越多，对学生成绩的管理也显得越来越复杂烦琐，因此使用学生成绩管理信息系统对成绩进行管理是十分必要的。通过使用数据管理系统，计算机可以代替人工进行数据录入、数据查询、数据更新、数据统计及报表统计、打印等功能，这样不仅可以降低出错率，还可以大大提高工作效率，并为以后的教学、考核等工作提供大量的前期数据，对提升教学管理质量也具有重大的指导意义。

本项目实施过程：

任务：设计“学生成绩管理信息系统”的主界面。

## 任务 设计“学生成绩管理信息系统”的主界面

### 任务目标

能设计并制作“学生成绩管理信息系统”的主界面。

能根据用户需要设计表单的属性。

### 任务描述

设计并实现“学生成绩管理信息系统”主表单，如图 5-1 所示。



## 相关知识

Visual FoxPro 9.0 不但支持传统的面向过程程序设计,而且在程序设计方面进行了扩展,提供了面向对象程序设计的强大功能。面向过程程序设计中所涉及的编程方法和编程技巧为面向对象程序设计奠定了良好的基础。在面向对象程序设计中,要考虑如何创建对象,利用对象来简化程序设计,并提供代码的可重用性。

Visual FoxPro 9.0 提供的程序设计功能和可视化的设计工具,使用户可以同时使用面向过程和面向对象的程序设计方法来提高应用程序的开发效率。



图 5-1 “学生成绩管理信息系统”主表单

### 1. 类和对象

类是对象的模板,对象是类的实例。按照抽象的模板创建出来的具体实例就是对象。根据一个类可以创建出多个具体的对象。例如,交通工具是类,而具体的汽车、火车、轮船等就是对象。Visual FoxPro 数据库中提供了一个表单类,通过它能创建出具体的表单对象(例如,后面创建的用于实现登录功能的表单 frmLogin 就是一个表单对象)。

### 2. 面向对象的三大特点

#### (1) 封装性

封装是指将对象的有关信息(数据和行为)隐藏起来,这样对象的使用者无须知道对象的设计者是如何设计对象的,进而达到保护代码不被破坏的目的。用户通过对象的接口就可以使用对象的属性和方法。

#### (2) 继承性

继承是指基于现有类创建新类时,新类将延用现有类的属性和方法。现有类称为父类,新类称为子类。这种继承机制具备自动更新的功能,即当对父类进行修改时,子类同时会接受到修改,如父类增加一个属性,子类也会增加相同的属性,当然子类也可以独立增加新的属性和方法,使其优于父类或进行个性化操作。

#### (3) 多态性

多态是指类中的方法会因为对象的不同而产生不同的结果。例如,灯具有发光的行为,但不同的灯发射出的光的颜色不同,节能灯为白色,彩灯颜色各不相同。

### 3. 对象的属性

Visual FoxPro 数据库中的表单(对象)具有名称、背景颜色、宽度、高度等性质,像这样的对象的特征称为对象的属性。例如,学生(对象)的属性有学号、姓名、性别、年龄等。

### 4. 对象的事件

对象的事件由用户或系统的操作来触发。例如,当单击表单右上角的“关闭”按钮时,将触发表单执行关闭行为,表单执行该行为时可能需要释放资源及进行一些相关的操作。事件是 Visual FoxPro 系统预先定义好的。对象执行事件的命令格式:对象.事件。

### 5. 对象的方法

方法是为了使对象实现特定功能而编写的程序代码,可以由用户创建和调用。对象调用方



法的命令格式：对象．方法。例如，为命令按钮的 Click 事件设置关闭表单的方法，当用户单击这个命令按钮时，系统激活按钮对象的 Click 事件，自动执行在该按钮对象中预先定义好的方法，完成关闭表单的操作。

对象的属性、事件和方法称为对象的三要素。

## 6. 容器类和控件类

Visual FoxPro 中的类可以分为容器类和控件类。容器类是多个对象的集合，控件类是单一的对象。Visual FoxPro 中常见的容器类有表单集、表格、页框、选项按钮组、命令按钮组等；常见的控件类有标签、文本框、命令按钮、编辑框、计时器、图像等。

## 7. 表单

表单（Form）是本任务中重点学习的对象，可以将各种图形、图像、数据等通过表单显示出来，为用户提供方便、友好的应用程序界面。表单文件的扩展名为“ .scx ”。

表单具有和 Windows 应用程序的窗口界面相似的结构特征，包括图标、标题、最大化按钮、最小化按钮、关闭按钮等。因为表单是一个对象，所以可以通过修改表单的属性来改变表单的结构特征，它也具有事件和方法。

表单的常用属性如表 5-1 所示。

表 5-1 表单的常用属性

| 属性名称        | 功能                                                | 默认值           |
|-------------|---------------------------------------------------|---------------|
| AutoCenter  | 控制表单初始化时是否总是位于 Visual FoxPro 窗口的中央                | 假（.F.）        |
| BackColor   | 控制表单的背景颜色                                         | 255, 255, 255 |
| BorderStyle | 控制表单边框的样式：无边框、单边框、固定边框和可调边框。如果值为 3，则用户能够自行改变表单的大小 | 3             |
| Caption     | 控制表单的标题显示的文字内容                                    | Form1         |
| Closable    | 控制表单标题栏中的关闭按钮是否可用                                 | 真（.T.）        |
| Movable     | 控制表单在运行时是否可以自由移动到新的位置                             | 真（.T.）        |
| MaxButton   | 控制表单在运行时是否有最大化按钮                                  | .T.           |
| MinButton   | 控制表单在运行时是否有最小化按钮                                  | .T.           |
| AlwaysOnTop | 控制表单在运行时是否一直保持在最前面                                | .F.           |
| Name        | 表单的名称                                             | Form1         |

表单的常用事件如表 5-2 所示。

表 5-2 表单的常用事件

| 事件名称     | 触发条件                                            | 触发方法         |
|----------|-------------------------------------------------|--------------|
| Load     | 当表单被装入内存时触发                                     | 表单名.load     |
| Init     | 当表单被创建时触发                                       | 表单名.init     |
| Activate | 当表单被激活时触发                                       | 表单名.activate |
| Unload   | 它是在释放表单之前发生的最后一个事件，它发生在 Destroy 事件和所有包含的对象被释放之后 | 表单名.unload   |
| Destroy  | 当表单被释放（关闭）时发生                                   | 表单名.destroy  |

表单的常用方法如表 5-3 所示。

表 5-3 表单的常用方法

| 方法名称    | 功能                            | 执行方法        |
|---------|-------------------------------|-------------|
| Hide    | 隐藏表单，通过设置 Visible 属性为“假”来隐藏表单 | 表单名.hide    |
| Show    | 显示表单                          | 表单名.show    |
| Release | 从内存中释放表单，即关闭表单                | 表单名.release |
| Refresh | 刷新表单，重新运行表单或者表单上的内容，并刷新所有的值   | 表单名.refresh |
| Quit    | 结束表单的运行，并退出 Visual FoxPro 系统  | 表单名.quit    |



实施步骤

1. 创建主表单

学生成绩管理信息系统主要包括 3 个模块：学生信息管理模块、课程信息管理模块和成绩信息管理模块，各模块应具有添加、删除、查询、修改和统计等功能。

选择项目管理器中的“文档”选项卡，选择“表单”选项，单击项目管理器右侧的“新建”按钮，在弹出的“新建表单”对话框中单击“新建表单”按钮，如图 5-2 所示。

单击工具栏中的“保存”按钮，保存表单名为 frmLogin，如图 5-3 所示。



图 5-2 新建表单



图 5-3 保存表单

2. 修改表单属性

参照表 5-4，在属性列表中修改表单属性。

表 5-4 frmLogin 表单的属性设置

| 属 性 名       | 属 性 值          |
|-------------|----------------|
| AlwaysOnTop | .T.            |
| AutoCenter  | .T.            |
| Caption     | 学生成绩管理信息系统登录界面 |
| Name        | frmLogin       |
| MaxButton   | .F.            |
| MinButton   | .F.            |
| MoveAble    | .F.            |

3. 运行表单

(1) 界面方式运行表单

- 在项目管理器中选中表单，单击右侧的“运行”按钮即可运行表单。



- 在项目管理器中选中表单，单击工具栏中的红色叹号按钮即可运行表单。

## (2) 命令方式运行表单

运行表单的命令如下：

```
DO FORM 表单名
```

在“命令”对话框中输入“DO FORM frmLogin”，即可运行表单。

## 4. 设计并运行主程序

选择项目管理器中的“代码”选项卡，新建程序，名为 main.prg，在其中输入运行表单的代码：

```
DO FORM frmLogin
```

运行主程序 main.prg，即可看到运行后的表单，如图 5-1 所示。



## 问题探究

### 1. 如何区分对象的事件与方法？

与对象的属性相比较，对象的事件和方法都是对象的动态行为。通常事件是被动行为，而方法是主动行为。

事件即对某个对象执行某些操作时会激活这个对象所具有的行为，这些行为都是 Visual Foxpro 系统为类预先定义好的。例如，单击时激活 Click 事件，释放表单所占资源时激活 Destroy 事件等，这些事件并不是用户编写的，这样的行为就是对象的事件。

方法是程序设计者因为某些特定的需求自己编写的一些代码而构成的程序，这样的程序是在遵循 Visual Foxpro 语法规则的基础上由程序员自己设计的，具有很大的灵活性。通常事件与方法是结合使用的。例如，在某按钮控件的 Click 事件的代码窗口输入如下命令：

```
USE Student
DO WHILE NOT EOF()
 DISPLAY
 SKIP
ENDDO
USE
```

则用户单击该命令按钮时就可以执行 Click 事件了，至于事件执行的内容则是由事件方法中的代码决定的。在此处，事件触发后要显示的是哪张表，如何显示该表中的记录则是由方法决定的。

### 2. 如何区分表单的名称、Name 属性和 Caption 属性？

表单的名称是指创建和保存表单文件时所指定的名称，其扩展名为“.scx”；Name 属性和 Caption 属性是在打开表单设计器后在属性列表中指定的，Name 属性代表表单的名称，是指在当前项目中使用表单的属性、事件、方法或调用表单时应使用的名称，如 frmLogin.Show()；Caption 属性是指表单标题栏显示的内容，代表表单中设计的内容，只起到显示的作用。

### 3. 如何理解 Visual Foxpro 面向对象程序设计与面向过程程序设计？

Visual Foxpro 与其他数据库管理系统或程序设计语言都不同，它既可以对数据库进行专业的管理，同时还能为用户提供友好的交互界面，使用面向过程、面向对象程序设计与数据库相结合，更加方便地处理数据库中的数据。

面向过程的程序设计主要应用顺序结构、选择结构和循环结构，在 Visual FoxPro 中，这 3 种结构对应不同于其他程序设计的命令语句。面向过程主要是以设计模块化的功能为过程，每

一个模块都能够实现一部分功能，再将各个模块组织起来以实现整个功能。而面向对象的程序设计主要采用对象和类的思想，其中应用其封装性、继承性、多态性的特性，将整个功能中的元素和行为都抽象为类，并为其设计属性、事件和方法，以客观存在的人或物或抽象概念为对象，从而应用其行为，以实现其功能。

典型的面向对象程序设计语言是 Java 语言，典型的面向过程程序设计语言是 C 语言。



## 阅读材料

### 1. 面向对象分析

面向对象（Object Oriented，OO）是一种软件开发方法，它是在结构化设计方法遇到了很多问题的情况下才产生的。

面向对象分析的目的是对客观存在的事物建立模型，以方便设计。

（1）面向对象的分析者通过与用户一起分析交流来提炼需求，因为只有这样才能真正了解并表示出用户的真实意图，这其中也涉及对需求的分析及对丢失信息的查找。

（2）建立对象模型时首先要确定标识和关联，因为它们关系着系统的整体结构和解决问题的方法，其次是增加属性，进一步详细地描述关联和类的基本关系网，并使用继承处理和组织类，最后把操作行为添加到类，构造出动态模型和功能模型。

（3）动态分析首先要寻找对象的事件，然后分析预测各个对象可能的事件顺序。对象的事件包括所有从用户来或到用户去的信息、外部设备传送的消息信号、输入、转换和动作，可能是常规事件，也可能是异常事件。对不同对象之间的事件建立排序表，并为各对象类建立状态图，以反映出各个对象所接收和发送的事件及其前面状态关系，每个事件都对应于状态图中一条路径。这是进行动态建模的过程。

（4）功能建模说明对象中的值是如何具体计算实现的，说明值之间的相互依赖关系等。要表示功能依赖关系主要采用数据流图，其中的处理应当与状态图相对应，其中的数据流对应于对象图中的对象或属性。

（5）确定操作。

### 2. 面向对象设计

面向对象设计是把分析阶段得到的结果转变成符合成本和质量要求的、抽象的系统实现方案的过程。从面向对象分析到面向对象设计，是一个逐渐扩充模型的过程，在这个过程中应遵循一定的面向对象设计准则。

模块化。面向对象开发方法是在面向过程的基础上得到的，面向对象设计中也包含了面向过程的程序设计，因此自然支持系统分解成模块的设计原则为在面向对象程序设计中对象就是模块。在这个模块中把数据结构和操作这些数据的方法紧密地结合在一起。分解系统是将系统分解成一组具有高内聚和低耦合的模块，这是面向对象模块化的属性。

抽象。抽象包括过程抽象和数据抽象。

信息隐藏。在面向对象方法中，信息隐藏通过对象的封装性来实现。

高内聚。模块内部（即对象内部）各元素间的联系应当是极其紧密的。

低耦合。低耦合主要指不同对象之间相互关联的紧密程度。

高内聚、低耦合是设计的一个重要标准，因为这有助于使得系统中某一部分的变化对其他



部分的影响降到最低程度,同时使得该模块内部是一个紧密完整的相对独立整体。



## 项目小结

本项目要求学生能够应用表单设计器根据实际需求创建一个表单,并能够对表单的相关属性进行设置,理解各个属性设置的含义,掌握使用方法。



## 上机实习

参照“学生成绩管理信息系统”的主界面设计“图书管理信息系统”的主界面,并参照“学生成绩管理信息系统”对“图书管理信息系统”的主界面表单的属性进行相应设置。

操作指导:建立并保存表单;设置表单属性并运行调试;通过程序运行、调用表单。



## 知识巩固

### 一、选择题

- Visual FoxPro 表单文件的扩展名为 ( )。
 

|         |         |         |         |
|---------|---------|---------|---------|
| A. .fpt | B. .mpr | C. .frx | D. .scx |
|---------|---------|---------|---------|
- 下列选项中 ( ) 是类。
 

|       |         |       |       |
|-------|---------|-------|-------|
| A. 汽车 | B. 交通工具 | C. 火车 | D. 轮船 |
|-------|---------|-------|-------|
- 下列 ( ) 属性表示表单显示的标题。
 

|         |         |          |            |
|---------|---------|----------|------------|
| A. Name | B. Size | C. Title | D. Caption |
|---------|---------|----------|------------|
- 关闭当前表单应使用的命令是 ( )。
 

|                     |                   |
|---------------------|-------------------|
| A. ThisForm.Release | B. ThisForm.Close |
| C. ThisForm.Quit    | D. ThisForm.Hide  |
- 能够实现信息隐藏的是 ( )。
 

|       |       |       |       |
|-------|-------|-------|-------|
| A. 封装 | B. 继承 | C. 多态 | D. 保护 |
|-------|-------|-------|-------|
- 能够实现信息共享的是 ( )。
 

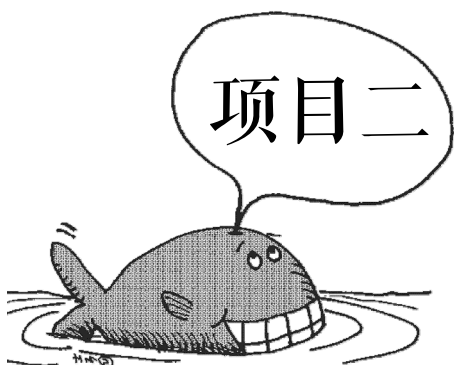
|       |       |       |       |
|-------|-------|-------|-------|
| A. 封装 | B. 继承 | C. 多态 | D. 保护 |
|-------|-------|-------|-------|
- 下列事件的发生顺序正确的是 ( )。
 

|                                |                                |
|--------------------------------|--------------------------------|
| A. Init, Load, Unload, Destroy | B. Init, Destroy, Load, Unload |
| C. Load, Init, Destroy, Unload | D. Load, Unload, Init, Destroy |
- 下列刷新表单的事件是 ( )。
 

|            |         |         |            |
|------------|---------|---------|------------|
| A. Release | B. Show | C. Hide | D. Refresh |
|------------|---------|---------|------------|

### 二、填空题

- 面向对象的三大特点是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- 每一个对象都包含\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_三要素。
- 表单的 AutoCenter 属性的含义是\_\_\_\_\_。
- 使用\_\_\_\_\_命令可以退出 Visual FoxPro 系统,返回操作系统。
- 要运行表单应使用的命令是\_\_\_\_\_。



# 设计并制作“学生成绩管理信息系统” 的登录界面

## 项目说明

为了保证系统数据的安全性，通常要为系统制作权限认证表单，即制作安全登录模块。方法是在表单中要求用户输入用户名和密码，如果系统用户表中存在该用户名，同时用户输入的密码与系统用户表中该用户的密码相同，则允许用户登录，进入系统主界面；否则，认为该用户是非法的，终止程序的执行。

本项目实施过程：

任务一：设计登录界面的背景图像。

任务二：设计登录界面的“用户名”和“密码”。

任务三：设计登录界面的“确定”和“取消”按钮。

## 任务一 设计登录界面的背景图像

### 任务目标

能为登录界面设置背景图像。

能根据背景图像的实际尺寸设置登录窗口的大小。

### 任务描述

使用图像控件设计登录表单的背景。



相关知识

图像控件（Image）用于创建图像对象。图像对象的作用是用在表单中显示图像信息。使用图像控件，在表单运行时只能显示图像，并不能编辑图像。图像控件可以在表单上显示各种格式的图像，如 BMP、JPG、GIF、ICON 等格式图像。图像控件还可以显示通用字段的内容，图像控件在工具栏的位置如图 5-4 所示。。

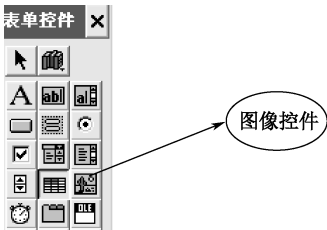


图 5-4 图像控件

图像控件常用的常用属性如表 5-5 所示。

表 5-5 图像控件的常用属性

| 属 性         | 功 能                        | 默 认 值 |
|-------------|----------------------------|-------|
| Picture     | 控制图像控件要显示的图像来源，属性值为图像文件的地址 | 无     |
| BackStyle   | 控制图像的背景色是否为透明              | 1-不透明 |
| Stretch     | 控制在调整图像尺寸大小时图像的变化状态        | 0-剪裁  |
| BorderStyle | 控制图像是否显示边框                 | 0-无边框 |
| Visible     | 控制图像是否是可见的                 | .T.   |

说明：Stretch 属性的值可以是 0-剪裁，1-等比填充，2-变比填充。其中 0-剪裁表示调整图像控件的大小，使其刚好可以显示完整的图片；1-等比填充表示放大或缩小图像以适应图像控件的大小，但要保持图像的原来比例；2-变比填充表示调整图像的大小以适应图像控件的大小，但不保持图像原来的比例。



实施步骤

1. 增加图像控件

从表单控件工具栏中选择图像控件，在表单上拖出适当大小。如图 5-5 所示。



图 5-5 拖放图像控件

2. 设置图像控件的属性

在属性列表中参照表 5-6 设置图像控件的属性。

表 5-6 frmLogin 表单中图像控件的属性

| 属 性 名   | 属 性 值  |
|---------|--------|
| Picture | 自选图像   |
| Stretch | 2-变比填充 |

3. 调整表单与图像大小

根据实际情况调整表单大小，可以发现因为设置了 Stretch 属性为 2-变比填充，图像可以调整大小，否则图像不可拖动改变尺寸。

任务二 设计登录界面的“用户名”和“密码”

任务目标

能为登录界面设置“用户名”输入框。



能为登录界面设置“密码”输入框。

任务描述

使用标签、文本框控件设计登录界面的用户名和密码。

相关知识

1. 标签控件

标签控件 ( Label ) 用于创建标签对象，如图 5-6 所示。标签的作用是用于显示文本信息，只要把要显示的文本直接输入到标签的 Caption 属性即可。用户可以通过标签自身提供的属性窗口，修改其 Caption 属性的值。需要注意的是，表单不仅能通过属性窗口修改标签控件的 Caption 属性值，而且能在运行时修改该属性值。

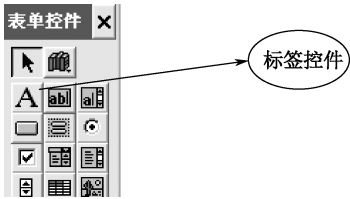


图 5-6 标签控件

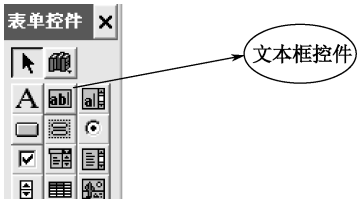
标签控件的常用属性如表 5-7 所示。

表 5-7 标签控件的常用属性

| 属 性         | 功 能                               | 默 认 值     |
|-------------|-----------------------------------|-----------|
| AutoSize    | 控制标签上文本信息的长度与标签的大小是否自动调整          | 假 ( .F. ) |
| Alignment   | 控制标签上文本信息的对齐方式，0-左对齐，1-右对齐，2-居中对齐 | 0         |
| BackColor   | 控制标签的背景颜色                         | 根据系统而不同   |
| BackStyle   | 控制标签的背景是否透明                       | 1-不透明     |
| BorderStyle | 控制标签的边框样式                         | 0-无边框     |
| Caption     | 控制标签上要显示的文本信息                     | Label1    |
| ForeColor   | 控制标签上文本的颜色，又称前景色                  | 0,0,0     |
| FontName    | 控制标签上文本的字体                        | 宋体        |
| FontSize    | 控制标签上文本的字号                        | 9         |
| Name        | 控制标签的名称，如在程序中为标签动态显示内容时需要使用       | Label1    |
| WordWrap    | 控制标签上的文本是否可以换行显示，即可以纵向多行显示        | 假 ( .F. ) |

2. 文本框控件

文本框控件 ( Text ) 用于创建文本框对象，是表单中最常用的控件之一，如图 5-7 所示。文本框不仅可以用于接收用户输入的数据，还可以用于显示文本信息，并对其值进行计算。文本框中字符的个数最大为 255。



在文本框中输入字符超出文本框的固定宽度时会自动换行显示，以“Enter”作为输入的结束标志，按“Esc”键取消当前输入。

文本框的常用属性如表 5-8 所示。

图 5-7 文本框控件



表 5-8 文本框的常用属性

| 属 性           | 功 能                                                     | 默 认 值 |
|---------------|---------------------------------------------------------|-------|
| ControlSource | 控制文本框显示信息的数据源，使文本框显示的数据与数据源中某个字段的数据绑定                   | 无     |
| InputMask     | 对输入数据格式进行控制                                             | 无     |
| Name          | 文本框控件的名称，动态设置值时需要使用                                     | Text1 |
| PasswordChar  | 设置文本框中数据输出或者输入时的代替字符，用于保护输入或输出的数据，如输入密码时，用其他符号代替输入的密码数据 | 无     |
| ReadOnly      | 控制文本框中的信息是否为只读                                          | 假（F.） |
| Value         | 控制文本框中显示的数据的值，可以是数值型、字符型、日期型或逻辑型                        | 无     |
| Text          | 文本框中的内容                                                 | 0     |



## 实施步骤

### 1. 添加标签控件

向表单 frmLogin 中添加两个标签控件（Label1 和 Label2）。在控件工具栏单击“标签控件”按钮，当鼠标指针放置在表单上方，光标变成十字形时，在表单的适当位置拖动出一个矩形框到合适大小即可释放鼠标。

### 2. 修改标签控件的属性

在属性表中参照表 5-9 修改标签的属性。

表 5-9 frmLogin 表单中标签控件的属性

| 属 性 名     | 属 性 值 |
|-----------|-------|
| AutoSize  | .T.   |
| BackStyle | 0-透明  |
| Caption   | 用户名   |
| FontBold  | .T.   |
| FontSize  | 16    |
| ForeColor | 自选    |

### 3. 添加文本框控件

在控件工具栏中单击“文本框控件”按钮，在表单上添加两个文本框控件，分别为 tName 和 tPassword。

### 4. 修改文本框控件属性

在属性表中参照表 5-10 修改文本框控件的属性。

表 5-10 frmLogin 表单中文本框控件的属性

| tName    |     | tPassword    |     |
|----------|-----|--------------|-----|
| 属性名      | 属性值 | 属性名          | 属性值 |
| FontBold | .T. | FontBold     | .T. |
| FontSize | 16  | FontSize     | 16  |
|          |     | PasswordChar | #   |

运行效果如图 5-8 所示。

### 任务三 设计登录界面的“确定”和“取消”按钮

#### 任务目标

- 能为登录界面设置“确定”按钮，使其实现登录系统的功能。
- 能为登录界面设置“取消”按钮，使其实现关闭系统的功能。

#### 任务描述

使用命令按钮控件设计并实现登录界面的功能。

#### 相关知识

##### 1. 命令按钮控件

命令按钮控件（Command）用于创建命令按钮对象，如图 5-9 所示。命令按钮的作用是用于控制程序的运行，如运行指定代码或表单，清空或保存编辑的内容，关闭或刷新表单内容等。

命令按钮的常用事件和属性如表 5-11 和表 5-12 所示。



图 5-8 运行效果

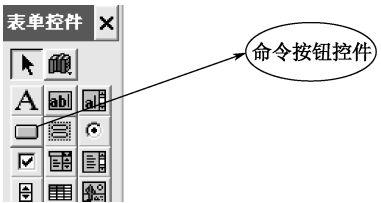


图 5-9 命令按钮控件

表 5-11 命令按钮的常用事件

| 事 件 名    | 功 能                         |
|----------|-----------------------------|
| Click    | 单击命令按钮时执行 Click 过程中设计的代码    |
| DblClick | 双击命令按钮时执行 DblClick 过程中设计的代码 |

表 5-12 命令按钮的常用属性

| 属 性 名   | 功 能                      | 默 认 值 |
|---------|--------------------------|-------|
| Caption | 控制命令按钮显示的内容              | 无     |
| Cancel  | 指定命令按钮是否为“取消”按钮          | .F.   |
| Default | 指定按 Enter 键时，哪一个命令按钮进行响应 | .F.   |
| Enabled | 控制命令按钮是否可用               | .T.   |
| Picture | 在命令按钮上显示图像               | 无     |



#### 实施步骤

##### 1. 添加“确定”按钮并修改其属性

在控件工具栏中单击命令按钮控件，当鼠标指针变成“十”字形时，在表单的适当位置拖



动出一个命令按钮。将按钮的 Caption 属性值改为“确定”,按钮的 Name 属性值改为“cmdOK”。

## 2. 添加“取消”按钮并修改其属性

在表单的适当位置添加取消按钮。将按钮的 Caption 属性值改为“取消”,按钮的 Name 属性值改为“cmdCancel”。

## 3. 建立用户信息表

使用表设计器新建表,表名为 DefaultSet。表结构如表 5-13 所示,表记录如表 5-14 所示。

表 5-13 DefaultSet 表的表结构

| 字段名 | 类型  | 宽度 |
|-----|-----|----|
| 名称  | 字符型 | 20 |
| 对应值 | 字符型 | 20 |

表 5-14 DefaultSet 表记录

| 名称            | 对应值             |
|---------------|-----------------|
| loginName     | sa              |
| loginPassword | 123456 (可以自行设定) |

## 4. 设计表单的初始化代码

要求运行表单时最大的密码容错次数为 3,并设置背景图片与窗口的大小相符。双击表单,在表单的 Init 事件中输入如下代码:

```
PUBLIC i
i=0
ThisForm.Image1.Left=0
ThisForm.Image1.Top=0
ThisForm.Image1.Width=ThisForm.Width
ThisForm.Image1.Height=ThisForm.Height
```

## 5. 设计“确定”按钮的代码

双击“确定”按钮,在其 Click 事件中输入如下代码:

```
&& 登录次数
i=i+1
IF i>3
 MessageBox("登录次数大于 3 次,拒绝访问",0+48+0,"提示信息")
&& 参见本项目阅读材料
ThisForm.Release
CLEAR EVENTS
CLOSE ALL
SET TALK ON
SET SAFETY ON
QUIT
ENDIF
&&验证用户名和密码
IF Used("DefaultSet")
 SELECT "DefaultSet"
```

```

ELSE
 SELECT 0
 USE DefaultSet
ENDIF
GO TOP
LOCATE FOR 名称= "LoginName "
IF Found() AND AllTrim(ThisForm.tName.Text)=AllTrim(对应值)
&& 以下验证密码
 GO TOP
 LOCATE FOR 名称= "LoginPassword "
 IF Found() AND Alltrim(ThisForm.tPassword.Text)=AllTrim(对应值)
 ThisForm.Release
 DO FORM frmMain
 &&用户名和密码均正确，执行程序主表单 frmMain, 该表单将在下一个项目中创建
 ELSE
 MessageBox(" 密码错误! ",0+48+0, " 提示信息 ")
 ThisForm.tPassword.SelStart=0
 &&参见本项目阅读材料
 ThisForm.tPassword.SelLength=Len(AllTrim(ThisForm.tName.Text))
 ThisForm.tPassword.SetFocus
 && 参见本项目阅读材料
 ENDIF
ELSE
 MessageBox(" 用户名错误! ",0+48+0, " 提示信息 ")
 ThisForm.tName.SelStart=0
 ThisForm.tName.SelLength=Len(AllTrim(ThisForm.tName.Text))
 ThisForm.tName . SetFocus
ENDIF
ENDIF

```

## 6. 设计“取消”按钮的代码

“取消”按钮的代码如下：

```

ThisForm.Release
SET TALK ON
SET SAFETY ON
SET DELETED ON
CLEAR ALL

```

## 7. 设计主程序

新建程序名为 main.prg，输入如下代码：

```

SET TALK OFF
SET DELETED OFF
SET SAFETY OFF
SET ESCAPE OFF
newPath=SYS(16)
newPath=getPath(newPath)
newPathApp=newPath+ " \app "
newPathImage=newPath+ " \Image "
newPathScx=newPath+ " \scx "
newPathTable=newPath+ " \table "
newPathPrg=newPath+ " \prg "
newPathMnt=newPath+ " \mnt "
&&获取程序全路径名

```



```

&&设置默认路径
SET PATH TO (newPath+";"+newPathApp+";"+newPathImage+";"+newPathscx+
";"+newPathTable+";"+newPathPrg+";"+newPathMnt)
&&打开数据库
OPEN DATABASE Student
&&执行登录表单
DO FORM frmLogin
&&获取从全路径右侧去掉文件名和最后一层目录的路径函数
FUNCTION getPath
PARAMETERS mPath
result=mPath
FOR i=1 TO 3
 rm=Rat('\',result)
 lm=At('\',result)
 IF rm=lm
 result=Substr(result,1,lm)
 ELSE
 result=Substr(result,1,rm-1)
 ENDIF
 i=i+1
ENDFOR
RETURN result
ENDFUNC

```



## 问题探究

### 1. 如何确定文本框所对应的表中字段？

Visual FoxPro 表单中的控件分为绑定型控件和非绑定型控件。所谓绑定型控件是指控件的内容与表中数据是相关的，非绑定型控件的值则与数据表无关。

文本框既可以是绑定型，也可以是非绑定型。将文本框设置为绑定型控件的方法一是添加控件后设置控件的 ControlSource 属性；二是将表中的字段直接拖动到表单中，这样会自动生成相应类型的控件。但无论使用哪种方法，前提条件是必须设置表单的数据环境，方法有如下两种。

在表单的空白处右击，在弹出的快捷菜单中选择“数据环境”命令并添加表。

在表单设计器中选择“显示”“数据环境项”命令并添加表。



## 阅读材料

### 1. Used() 函数

Used() 函数用于判断一个表是否打开。若函数的返回值是“.T.”，则表示已经打开，可以直接使用；若返回值是“.F.”，表示没有打开，需要使用显式打开命令“USE 表名”打开指定的表。

Used() 函数通常与 If 语句结合使用。

### 2. SelStart 属性

编辑框的 SelStart 属性表示从编辑框中选取文本的起始位置。如果从第一个字符开始选，则 SelStart 值应为 0，如果从第二个字符开始选，则 SelStart 值应为 1，……如果从第 n 个字符

开始选, 则 SelStart 值应为  $n-1$ 。

例如:

```
n=At("good", ThisForm.Edit1.Value)
IF n<>0
 ThisForm.Edit1.SelStart=n-1
 ThisForm.Edit1.SelLength=Len("good")
 && SelLength:选取文本的长度, 即选取字符的位数
ENDIF
```

在编辑框输入 “It is a good book”, 再执行上述一段程序代码, 执行结果是字符串 “good” 被选中。

字符串 “good” 在 “It is a good book” 字符串中的实际起始位置应该是 9, 而上述代码中第三行表示选取的起始位置是 8, 即  $n-1$ 。

### 3. 文本框和编辑框 (Edit) 控件的 Text 属性与 Value 属性

Text 属性与 Value 属性都可以表示文本框的内容, 但 Text 属性中包含的值是没有格式限制的, 且与用户在控件中输入的文本是完全相同的。

而 Value 属性表示文本框中输入的可能不是字符型, 它可能包含日期型或数值型等其他类型的内容。

对于编辑框 (Edit) 控件的使用方法将在本单元项目四中介绍。

### 4. SetFocus 方法

SetFocus 方法是指使某个控件获得焦点。例如, ThisForm.Text1.SetFocus 表示 Text1 文本框获得焦点, 一旦控件获得了焦点, 用户的所有输入都针对这个控件。例如, 登录界面可以使登录的用户名文本框在运行表单时首先获得焦点, 这样用户不需要使用鼠标选中文本框即可率先输入用户名。

### 5. Show 方法

按照模式或无模式属性显示一个表单, 如 ThisForm.Show()。模式表单在显示时, 该表单一直占有显示焦点, 其他表单不能被激活, 而无模式表单显示时, 控制焦点可以切换到其他表单上。表单的模式属性可以使用 WindowType 属性设置。

### 6. MessageBox() 函数

MessageBox() 函数用来显示一个对话框, 该对话框中可以由用户自己定义要显示的内容、按钮及图标。

#### (1) 语法格式

```
MessageBox (提示信息[, 对话框的属型[, 对话框窗口标题]])
```

#### (2) 参数设置

- 提示信息: 指定对话框中显示的文本。在该提示信息中如需换行显示, 需要使用回车符 (CHR(13))。对话框的高度和宽度会根据输入的提示信息内容自动调整大小, 以显示全部信息。

- 对话框的属型: 对话框的属型最多由 3 个数字通过 “+” 相连接, 分别指定对话框中显示的按钮类型、按钮图标及设置好的默认按钮。例如, 在属型参数 “2+32+256” 中, 2 表示对话框中包含的按钮为 “放弃”、“重试”、“忽略” 3 个; 32 表示对话框中显示的图标为问号; 256



表示按 Enter 键时，默认执行的按钮是第二个按钮，即“重试”。也可以只指定 3 项中的任意一项或两项；当省略本参数时，等同于指定其值为 0。参数设置列表如表 5-15 所示。

表 5-15 MessageBox 的参数设置列表

| 设置按钮属性     | 按钮图标       | 默认按钮      |
|------------|------------|-----------|
| 0：确定       | 16：“停止”图标  | 0：第一个按钮   |
| 1：确定、取消    | 32：问号      | 256：第二个按钮 |
| 2：放弃、重试、忽略 | 48：惊叹号     | 512：第三个按钮 |
| 3：是、否、取消   | 64：信息(i)图标 |           |
| 4：是、否      |            |           |
| 5：重试、取消    |            |           |

### (3) 返回值

MessageBox()函数的返回值类型为数值型。MessageBox()的返回值表示用户选择了对话框中的哪个按钮。在含有“取消”按钮的对话框中，如果按 Esc 键退出对话框，则与单击“取消”按钮一样，返回值为 2。

MessageBox()函数的返回值列表如表 5-16 所示。

表 5-16 MessageBox 函数的返回值列表

| 返回值 | 按钮 |
|-----|----|
| 1   | 确定 |
| 2   | 取消 |
| 3   | 放弃 |
| 4   | 重试 |
| 5   | 忽略 |
| 6   | 是  |
| 7   | 否  |

例如：

```
If MessageBox("确定要删除?", 1+32+256, "确认退出")=1
 PACK
ENDIF
```



## 项目小结

使用 Visual FoxPro 设计数据库应用系统，通常都会在登录界面设计用户名与密码的验证信息。本项目以学生成绩管理信息系统为例，介绍了用户登录界面设计的全过程，其中包括表的设计、表单界面的设计、控件属性及方法的编写，为其他应用系统的设计提供了良好的参考。



## 上机实习

参照“学生成绩管理信息系统”制作“图书管理系统”的登录界面，如图 5-10 所示。

操作指导：设计表单的背景图像；为表单添加控件；设置各个控件属性；设计“确定”和“取消”按钮的命令代码；运行并调试表单。





图 5-10 “图书管理系统”登录界面



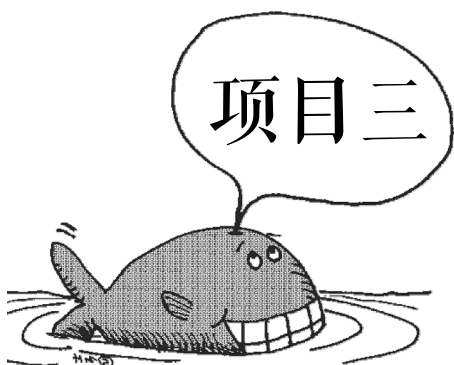
## 知识巩固

### 一、选择题

1. 主要用来显示文本内容的控件是 ( )。  
A. 文本框      B. 标签      C. 命令按钮      D. 表格
2. 下列 ( ) 控件最适宜制作表单的背景。  
A. 图像      B. 文本框      C. 标签      D. OLE
3. 使得标签或文本框内容根据控件大小自动调整大小的属性是 ( )。  
A. AutoCenter      B. AutoSize  
C. Size      D. Left, Right, Top, Bottom
4. 表示标签显示内容的属性是 ( )。  
A. Value      B. Text      C. Name      D. Caption
5. 单击命令按钮时激活该控件的 ( ) 事件。  
A. Click      B. DBLClick      C. RightClick      D. Release
6. 一个表单的多个命令按钮中能有 ( ) 按钮的 Default 属性值为 “.T.”。  
A. 一个      B. 两个      C. 3 个      D. 任意个
7. 当一个表单中某个命令按钮的 ( ) 属性为 “.T.” 时, 表示按 Esc 键时执行该按钮的 Click 事件的过程代码。  
A. Cancel      B. Default      C. Enabled      D. Caption
8. 下列选项中 ( ) 不能表示变量的作用范围。  
A. Public      B. Private      C. Local      D. Personal

### 二、填空题

1. 图像控件的\_\_\_\_\_属性表示图像来源。
2. 要使用文本框显示密码为 “\*”, 则应设置文本框的\_\_\_\_\_属性值为 “\*”。
3. 在文本框中输入\_\_\_\_\_时表示输入结束, \_\_\_\_\_表示取消输入。
4. 要将文本框中的 txtName 值设置为 Student 表的 “姓名” 字段值时, 应使用的命令是\_\_\_\_\_。
5. 设置公共变量应使用的命令是\_\_\_\_\_。



## 创建“学生成绩管理系统” 主表单菜单

### 项目说明

就像 Windows 操作系统中的菜单一样，Visual FoxPro 设计的菜单也分为两种：下拉菜单和快捷菜单。下拉菜单包括菜单标题和菜单选项，其中菜单选项的功能可以通过命令、过程或者子菜单来实现。下拉菜单在运行时会出现在应用程序窗口顶部的菜单栏中，快捷菜单是右击时弹出的菜单。

“学生成绩管理系统”的所有功能都应在主表单的菜单中体现，本系统的菜单构成如图 5-11 所示。

本项目实施过程：

任务：设计“学生成绩管理系统”的主表单菜单。

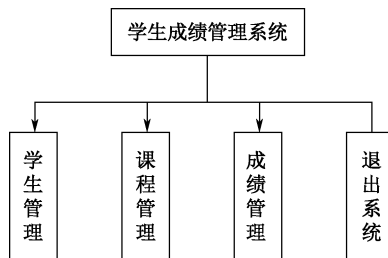


图 5-11 “学生成绩管理信息系统”的菜单构成

### 任务 设计“学生成绩管理系统”的主表单菜单

#### 任务目标

能为“学生成绩管理系统”的主表单设计菜单。

#### 任务描述

创建“学生成绩管理系统”主表单的菜单，效果如图 5-12 所示。

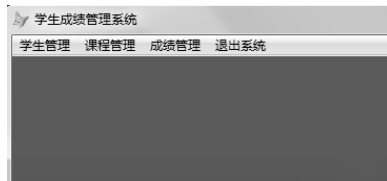


图 5-12 应用菜单效果图

相关知识

1. 创建下拉菜单

要建立一个下拉菜单，可以采用以下方法。

(1) 使用项目管理器创建下拉菜单

打开项目管理器，在“其他”选项卡中选择“菜单”选项，单击“新建”按钮，在弹出的“新建菜单”对话框中单击“菜单”按钮，即可打开菜单设计器进行菜单设计，如图 5-13 所示。

(2) 使用 Visual FoxPro 系统菜单项创建用户自定义下拉菜单

选择“文件” “新建”命令，在弹出的“新建”对话框中选中“菜单”单选按钮，可以打开菜单设计器，如图 5-14 所示。



图 5-13 “新建菜单”对话框

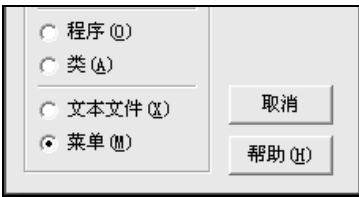


图 5-14 “新建”对话框

还可以使用命令创建菜单。

2. 生成菜单程序

若要运行新建的菜单文件(扩展名为“.mnx”),必须先将菜单文件生成菜单程序文件(即生成扩展名为“.mpr”的文件),生成步骤如下。

在菜单设计器环境下，单击“生成”按钮，如图 5-15 所示。

在“生成菜单”对话框中指定菜单程序文件的名称及位置。

单击“生成”按钮。

3. 运行菜单程序

可以使用如下命令运行菜单：

```
do 菜单程序文件.mpr
```

运行菜单程序文件时，扩展名“.mpr”是不可省略的；通常在表单的方法中运行菜单程序。

4. 编辑菜单

与创建菜单相似，修改菜单也有多种方法，它们分别是使用项目管理器、使用系统菜单和使用命令方式修改。

(1) 使用项目管理器修改菜单

打开项目管理器，选择“其他”选项卡中的“菜单”选项，选中要修改的菜单文件，单击



“修改”按钮，即可打开菜单设计器对其进行修改。若要对子菜单进行修改，单击“编辑”按钮，如图 5-16 所示。

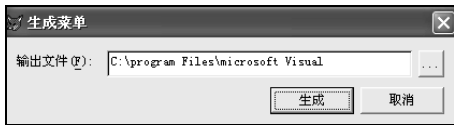


图 5-15 生成菜单



图 5-16 修改菜单

## (2) 使用 Visual FoxPro 系统菜单项修改用户自定义菜单

选择“文件”“打开”命令，在弹出的“打开”对话框中选择文件类型为“菜单”(.mnx 文件)，打开菜单设计器，即可对用户自定义菜单进行修改。

## (3) 使用命令方式修改菜单

使用命令方式修改菜单参见本项目阅读材料。



## 实施步骤

### 1. 建立主菜单

选择项目管理器中的“其他”选项卡，选择“菜单”选项，单击“新建”按钮，在弹出的“新建菜单”对话框中单击“菜单”按钮即可打开菜单设计器，输入菜单内容，如图 5-17 所示。

### 2. 建立“学生管理”子菜单

单击“学生管理”子菜单右侧的“创建”按钮，打开子菜单设计器，分别输入“学生管理”子菜单内容，如图 5-18 所示。

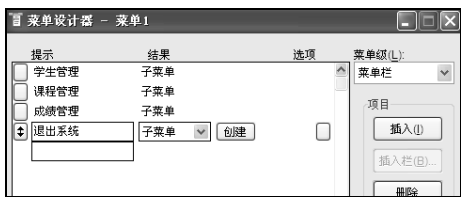


图 5-17 建立主菜单



图 5-18 建立“学生管理”子菜单

### 3. 建立“课程管理”子菜单

单击“课程管理”子菜单右侧的“创建”按钮，打开子菜单设计器，分别输入“课程管理”子菜单内容，如图 5-19 所示。

### 4. 建立“成绩管理”子菜单

单击“成绩管理”子菜单右侧的“创建”按钮，打开子菜单设计器，分别输入“成绩管理”子菜单内容，如图 5-20 所示。

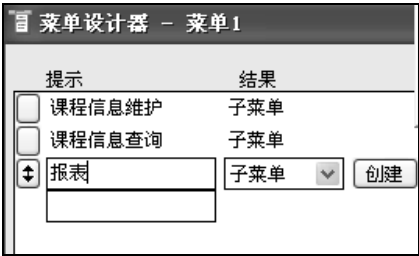


图 5-2 建立“课程管理”子菜单

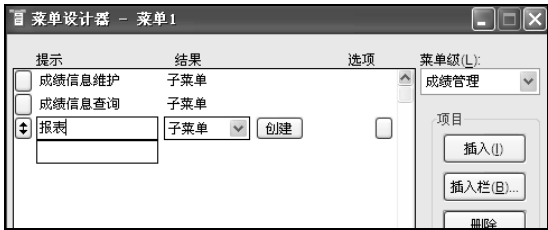


图 5-20 建立“成绩管理”子菜单

5. 保存、生成并运行菜单程序

单击“生成”按钮，弹出“保存”对话框，选择文件保存路径及文件名 Caidan.mnx，生成菜单程序 Caidan.mpr，再单击项目管理器右侧的“运行”按钮，执行效果如图 5-21 所示。

注：在项目管理器下直接运行菜单，系统会自动在用户定义的菜单项后添加显示“项目”菜单项，其中包括文件的新建、添加、修改等命令，同时系统原有工具栏仍然会显示，但不能使用；如果将菜单应用于表单，则不会产生附加的“项目”及工具栏等内容。

6. 将菜单应用于指定表单

在菜单设计器中打开 Caidan.mnx，选择“显示”“常规选项”命令，在弹出的“常规选项”对话框中“顶层表单”复选框，使得菜单程序应用于顶层表单，如图 5-22 所示。



图 5-21 生成并运行菜单

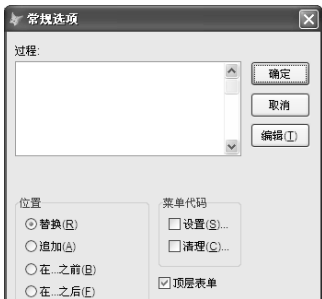


图 5-22 将菜单应用于顶层表单

新建表单 frmMain 作为主表单，表单属性如表 5-17 所示。

表 5-17 frmMain 表单的属性设置

| 属 性 名       | 属 性 值    |
|-------------|----------|
| AutoCenter  | .T.      |
| Caption     | 学生成绩管理系统 |
| Name        | frmMain  |
| MaxButton   | .T.      |
| MinButton   | .T.      |
| MoveAble    | .F.      |
| ShowWindow  | 2-作为顶层表单 |
| WindowState | 0-普通     |

在 frmMain 的 Init 事件中添加的程序代码如下：

```
DO Caidan.mpr WITH This ,.T.
```



执行后的效果如图 5-15 所示。

## 7. 设计“退出系统”命令

打开菜单设计器，在退出系统的操作过程中选择命令，输入 QUIT 命令即可退出 Visual FoxPro 系统，如图 5-23 所示。



## 问题探究

1. 如何确定何时使用界面方式或命令方式创建数据库对象？

创建菜单或其他数据库对象通常都有两种方式，即界面方式和命令方式。如果需要在程序执行过程中创建，则可以使用命令方式，其他情况可以在设计时使用界面方式创建。但在创建菜单时通常使用界面方式，因为一般菜单都是实现系统功能的导航，用户通过调用菜单执行其对应的程序或命令。

2. 如何区分下拉菜单和快捷菜单？

下拉菜单运行时位于表单窗口的上方菜单栏位置，当单击展开或使用相应快捷键展开时，可以继续选择其下一级子菜单；而快捷菜单是通过在某个对象（表单或控件）上右击才打开的菜单。

3. 如何区分菜单的热键与快捷键？

热键是指菜单名右侧括号中带下画线的字母，如“文件(E)”，使用时需要使用控制键 Alt+括号中对应的字母方能激活其对应的菜单功能。

快捷键通常是下拉菜单展开后选项中对应的辅助功能键（如 Ctrl、Alt 等）与字母的组合键，如文件菜单中“打开”后“Ctrl+O”就是打开的快捷键。使用快捷键时不需要展开其所在的菜单，而直接按相应的快捷键即可。



图 5-23 “退出系统”菜单



## 阅读材料

### 1. 快捷菜单

#### (1) 创建快捷菜单

创建快捷菜单的方法与创建下拉菜单的方法相似，可以使用菜单、项目管理器或使用 CREATE MENU 命令方式打开“新建菜单”对话框，单击“快捷菜单”按钮即可，如图 5-24 所示。

在弹出的快捷菜单设计器中对快捷菜单进行设计，如图 5-25 所示。可以参照下拉菜单设计器的使用方法来使用快捷菜单设计器创建快捷菜单。

#### (2) 设置热键

若要打开 Visual FoxPro 系统的“文件(E)”菜单，可以按 Alt+F 组合键，这里的字母“F”称为“文件(E)”菜单的热键。设置热键只需要打开菜单设计器，在“菜单名称”选项组中相应位置添加热键的字母，并在其前输入符号“\<”即可。例如，“编辑(E)”菜单项的热键为“E”，其菜单名称写成“编辑(\<E)”，在运行时会自动在热键下方加下画线，即变成“编辑(E)”，使用热键时，只需要同时按 Alt 键和 F 键，即可打开编辑菜单。



图 5-24 “新建菜单”对话框

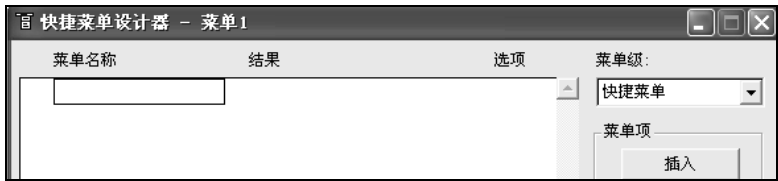


图 5-25 快捷菜单设计器

(3) 设置快捷键

对于某些常用的菜单命令，可以使用快捷键直接应用。通常同时使用 Ctrl 键和指定字母的组合作为快捷键来执行相应的菜单命令。如果用户熟悉了常用菜单命令的快捷键，那么不需要打开菜单，即可直接执行菜单命令，非常方便快捷。设置快捷键时，只需要打开菜单设计器，在“菜单名称”选项组中选中要设置快捷键的菜单名称，单击“选项”按钮，弹出“提示选项”对话框，在“键标签”和“键说明”文本框中输入相应的快捷键组合（输入时只需要同时按两个键，就会自动添加到文本框中）。例如，将“复制”的快捷键设置为 Ctrl+C，如图 5-26 所示。

2. 使用命令创建下拉菜单

创建菜单可以使用命令方式，格式如下：

CREATE MENU 菜单文件名

使用命令新建菜单文件时不需要指定扩展名，其默认的扩展名是“.mnx”；如果省略菜单文件名，系统会自动弹出“保存”对话框，要求填写菜单文件名及保存位置。

3. 快速创建菜单

有一些菜单功能是大多数应用程序都需要的功能，如“打开”、“复制”等，因此 Visual FoxPro 系统为用户提供了一些预先定义好的菜单，这些菜单具有 Windows 系统菜单的风格，各个菜单的基本功能已经设置好，用户可以在此基础上进行修改，这种设计方法操作更加简单，速度更快，为用户提供了极大的方便。

打开菜单设计器，在 Visual FoxPro 系统主菜单中选择“菜单”“快速菜单”命令，会自动生成与 Visual FoxPro 系统菜单相似的快速菜单，如图 5-27 所示。



图 5-26 快捷键的设置

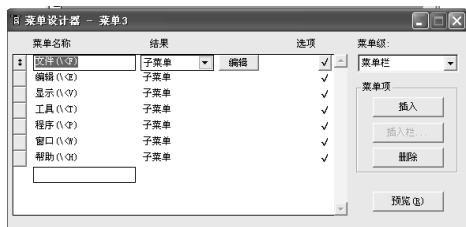


图 5-23 快速创建菜单

说明：

快速菜单只能创建下拉菜单，不能创建快捷菜单。

快速菜单只能在新建菜单设计器打开后直接使用，不能在设计了其他菜单选项后再使用。用户可以对系统提供的快速菜单进行修改、添加或删除，只需要通过右侧的按钮就可



以实现。

#### 4. 命令方式修改菜单

修改菜单的命令如下：

MODIFY MENU 菜单文件名

若省略菜单文件名，系统会自动弹出“打开”对话框，要求选择要修改的菜单文件。



### 项目小结

每一个应用系统都能够实现若干功能，而将这些功能系统地展现在用户面前，能够方便用户使用的导航功能可以通过菜单实现。本项目正是以“学生成绩管理系统”为例介绍菜单（主要是下拉菜单）的设计与实现方法，读者可以通过本项目掌握菜单的创建、编辑、生成、运行及应用于表单的方法。



### 上机实习

参照“学生成绩管理系统”实现“图书管理信息系统”的菜单设计。

1. 制作主菜单，包括图书管理、读者管理、退出系统 3 个主菜单项。
2. 制作图书管理子菜单，包括图书维护、图书管理、图书查询、借书报表 4 个子菜单项。
3. 制作读者管理子菜单，包括读者管理、读者添加两个子菜单项。
4. 保存菜单文件为 Book.mnx，并生成 Book.mpr 菜单程序文件。
5. 在“图书管理信息系统”的主界面表单中应用该菜单。



### 知识巩固

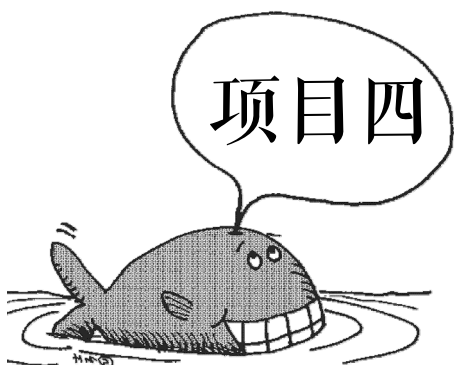
#### 一、选择题

1. 菜单文件和菜单程序文件的扩展名分别为 ( ) 和 ( )。  
A. .mnx                      B. .frx                      C. .qpr                      D. .mpr
2. 若菜单为“编辑(E)”，则下列 ( ) 是打开编辑菜单的热键。  
A. E                          B. Ctrl+E                      C. Alt+E                      D. Shift+E
3. 表单的 ( ) 属性表示可以将其设置为顶层表单。  
A. ShowWindow                      B. Top  
C. TopWindow                      D. Window
4. 可以按 Ctrl+C 实现“复制”菜单的功能，则 Ctrl+C 称为“复制”菜单的 ( )。  
A. 热键                      B. 快捷键                      C. 快速键                      D. 复制键

#### 二、填空题

1. 建立菜单后，必须\_\_\_\_\_才能运行菜单。
2. 要将下拉菜单应用于登录表单，需要将登录表单设置为\_\_\_\_\_。
3. 通过右击弹出的菜单称为\_\_\_\_\_。
4. 运行菜单的命令是\_\_\_\_\_。





## 实现学生成绩信息维护和查询功能

### 项目说明

一个系统往往需要执行多个相对独立的功能，因此一个应用程序往往由多个表单组成。在系统中通常通过主表单来对各个功能模块进行集中统一的管理和调用。学生成绩管理信息系统的主表单主要用于显示菜单，并将调用菜单所执行的表单都显示在主表单之上，从而实现系统界面的统一性。

本项目实施过程：

任务一：实现“学生信息维护”功能。

任务二：实现“学生信息查询”功能。

### 任务一 实现“学生信息维护”功能

#### 任务目标

能够制作“学生管理”菜单下“学生信息维护”子菜单项对应的功能。

具体内容包括：添加记录；修改记录；删除记录；关闭维护表单。

#### 任务描述

使用图形、表格、命令按钮组、选项按钮组、编辑框、列表框、组合框、计时器等控件设计并实现“学生信息维护”表单功能，效果如图 5-28 所示。



| 学号     | 姓名  | 出生日期     | 性别 | 专业    | 籍贯  | 照片地址 |
|--------|-----|----------|----|-------|-----|------|
| 120101 | 杜嘉  | 03/03/98 | 男  | 计算机应用 | 辽宁  | Gen  |
| 120102 | 王晓明 | 11/07/98 | 男  | 计算机应用 | 黑龙江 | Gen  |
| 120103 | 彭新信 | 05/22/99 | 女  | 计算机应用 | 辽宁  | gen  |
| 120104 | 李丽  | 08/17/99 | 女  | 计算机应用 | 山东  | Gen  |
| 120105 | 顾晓雪 | 04/27/98 | 女  | 计算机应用 | 河北  | Gen  |
| 120201 | 金明  | 02/09/99 | 男  | 电子技术  | 吉林  | Gen  |
| 120202 | 唐莎  | 10/10/98 | 女  | 电子技术  | 吉林  | Gen  |
| 120203 | 王敏敏 | 01/20/99 | 女  | 电子技术  | 黑龙江 | Gen  |
| 120204 | 李楠  | 07/08/98 | 女  | 电子技术  | 山东  | gen  |
| 120205 | 杨建  | 06/30/99 | 男  | 电子技术  | 河北  | gen  |

图 5-28 学生成绩管理信息系统的 Student 表格

相关知识

1. 图形控件

图形控件（Shape）也称为形状控件，属于输出型控件，用于创建形状对象，如图 5-29 所示。图形控件的作用是用于在表单上画出各种形状，如矩形、正方形、椭圆、圆等。图形控件的常用属性如表 5-18 所示。

表 5-18 图形控件的常用属性

| 属 性           | 功 能                                                     | 默 认 值   |
|---------------|---------------------------------------------------------|---------|
| Curvature     | 控制形状的弯曲度，值的范围为 0 ~ 99                                   | 0       |
| FillStyle     | 控制形状的填充方案，可以取值 0 ~ 7，分别代表不同的填充形式。各个取值的含义请参照控件的属性对话框中的描述 | 1-透明    |
| FillColor     | 控制形状的填充颜色                                               | 0, 0, 0 |
| SpecialEffect | 控制形状的特殊效果                                               | 1-平面    |

2. 表格控件

表格控件（Grid）用于创建表格对象，如图 5-30 所示。表格控件的作用是在表单中显示和处理数据表中的数据。表格控件的运行效果类似于表的浏览窗口，能够在表单中直接浏览或修改表中数据，使用方便。

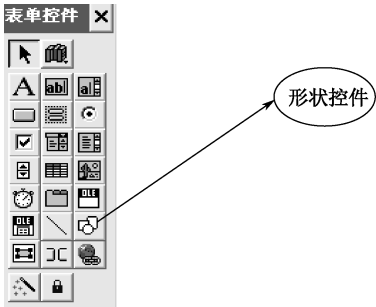


图 5-29 图形控件

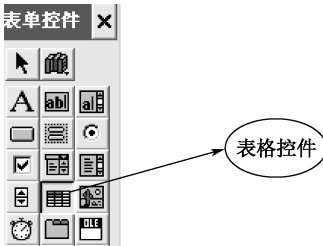


图 5-30 表格控件

使用表格时需要先设置表格要显示的数据源，因此需要先配置表单的数据环境。表格控件的常用属性如表 5-19 所示。

表 5-19 表格控件的常用属性

| 属 性              | 功 能            | 默 认 值 |
|------------------|----------------|-------|
| ColumnCount      | 设置表格中数据显示的列数   | 2     |
| RecordSource     | 设置表格中显示的数据源    | 0-无   |
| RecordSourceType | 设置表格中显示的数据源的类型 | 0-无   |

3. 命令按钮组控件

命令按钮组控件（CommandGroup）用于创建命令按钮组对象，如图 5-31 所示。命令按钮组控件的作用是用于产生一组命令按钮，每个按钮的功能与前面介绍的命令按钮相似，同时也可以将整个命令按钮组作为一个整体进行操作。当多个命令按钮用于实现相似的功能时，我们可以使用命令按钮组来操作。使用命令按钮组可以对代码进行重用，提高程序的运行效率。

命令按钮组控件的常用属性如表 5-20 所示。

表 5-19 命令按钮组控件的常用属性

| 属 性         | 功 能                              | 默 认 值 |
|-------------|----------------------------------|-------|
| ButtonCount | 设置命令按钮组中命令按钮的个数                  | 2     |
| Value       | 设置命令按钮组中被选择的命令按钮，值为当前被选择的命令按钮的序号 | 1     |

由于命令按钮组控件包含多个命令按钮，因此在单击时，系统会根据用户的单击位置来判断用户选择的是整个组还是单独的某一个命令按钮。当用户单击命令按钮组内的空白位置时，系统会执行整个命令按钮组的 Click 事件中对应的代码；当用户单击了某一个命令按钮时，系统会执行该命令按钮的 Click 事件中对应的代码。

4. 选项按钮组控件

选项按钮组控件（OptionGroup）用于创建选项按钮组对象，如图 5-32 所示。选项按钮组控件包含多个选项按钮，用户只能从多个选项按钮中选择一个选项，又称为单选按钮。选项按钮组控件的作用是提供多个单选按钮，用户可以通过单选按钮执行其对应的代码，实现指定的功能。当用户选中一个选项按钮时，其他选项按钮都会处于未选中状态。例如，某个选择题是单项选择题，可以使用表单和选项按钮组控件来实现。

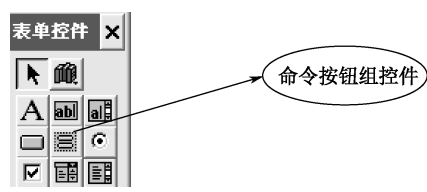


图 5-31 命令按钮组控件

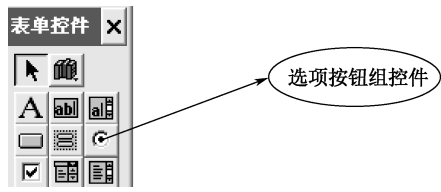


图 5-32 选项按钮组控件

选项按钮（Option）和选项按钮组（OptionGroup）控件的常用属性和事件如表 5-21 和表 5-22 所示。



表 5-20 选项按钮和选项按钮组控件的常用属性

| 控 件 类 型                | 属 性           | 功 能                                                               | 默 认 值            |
|------------------------|---------------|-------------------------------------------------------------------|------------------|
| 选项按钮<br>(Option)       | Alignment     | 设置选项按钮显示文本与控件按钮的位置                                                | 0-中部左对齐          |
|                        | Style         | 设置选项按钮的外观风格,值为0表示标准,值为1表示图形                                       | 0                |
|                        | Caption       | 设置各个选项按钮显示的文本信息                                                   | Option1, Option2 |
| 选项按钮组<br>(OptionGroup) | ButtonCount   | 选项按钮组中选项按钮的个数                                                     | 2                |
|                        | ControlSource | 设置与选项按钮组建立关联的数据源                                                  | 无                |
|                        | Value         | 设置选项按钮组中某一个选项按钮为默认选中的按钮,值为1表示被选中,值为0表示没有被选中,多个选项按钮时,只能有一个Value值为1 | 第1个为1,其他为0       |

表 5-21 选项按钮及选项按钮组的常用事件

| 事 件      | 触 发 条 件          |
|----------|------------------|
| Click    | 单击时触发            |
| DblClick | 双击时触发(用于选项按钮组控件) |

## 5. 编辑框控件

编辑框控件(Editbox)用于创建编辑框对象,如图5-33所示。编辑框控件与文本框控件相似,可以用于接收数据的输入或显示输出数据。与文本框控件不同的是,编辑框控件只能接收文本信息,而且可以接收多段文本信息,按Enter键时表示换行,而不像文本框控件一样表示结束输入。如果设定的文本框不能完全显示多段文本信息,系统会自动为其设置滚动条,以显示更多的文本。

编辑框控件一般用于显示或编辑较长的字符型字段或是备注型字段的内容。

编辑框控件的常用属性如表5-23所示。

表 5-22 编辑框控件的常用属性

| 属 性           | 功 能                    | 默 认 值 |
|---------------|------------------------|-------|
| ScrollBars    | 设置编辑框是否有滚动条            | 2-垂直  |
| ReadOnly      | 设置编辑框是否是只读的            | .F.   |
| ControlSource | 设置编辑框的数据源,使其显示指定数据源的数据 | 无     |
| Name          | 指定在代码中用以引用对象的名称        | Edit1 |
| Value         | 设置编辑框中显示的文本信息          | 无     |

## 6. 列表框控件

列表框控件(ListBox)用于创建列表框对象,如图5-34所示。列表框控件的作用是在运行表单时显示一个或者多个数据项,用户可以选择一个或多个数据项。列表框控件只允许用户选择数据项,不允许用户输入数据。

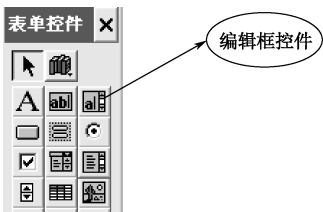


图 5-33 编辑框控件



图 5-34 列表框控件

列表框控件的常用属性如表 5-24 所示。

表 5-23 列表框控件的常用属性

| 属 性           | 功 能                                                                                                       | 默 认 值   |
|---------------|-----------------------------------------------------------------------------------------------------------|---------|
| ControlSource | 指定与列表框绑定的数据源。一般先将列表框中的值保存在数据源指定的字段内，配置好表单的数据环境后，将列表框与该数据源绑定，即可读出指定字段中存储的一组数据。与文本框控件的 ControlSource 属性含义不同 | 无       |
| MoveBars      | 控制是否在列表框中显示滚动条                                                                                            | .F.     |
| MultiSelect   | 控制是否能够在列表框中同时选择多行，按 Ctrl 键的同时单击，可以选择多个不连续的列表项；按 Shift 键的同时单击，可以选择多个连续的列表项                                 | 0 或 .F. |
| RowSource     | 指定列表框中显示的列表项的数据来源，通常为表中字段，需要先设置 RowSourceType 的值                                                          | 无       |
| RowSourceType | 指定列表框中数据源的类型，可以是值、表、查询语句、视图、数组等                                                                           | 0-无     |
| Selected      | 显示列表框中的列表项是否被选中                                                                                           | .F.为默认值 |

7. 组合框控件

组合框控件 (ComboBox) 用于创建组合框对象，如图 5-35 所示。组合框的作用与列表框类似，也可以显示多个数据项供用户选择。与列表框不同的是，组合框并不是把所有的数据项全部显示出来，而只显示其中一个数据项，这样可以节省表单的界面空间，这样的组合框又称为下拉列表框。

组合框还可以设置为下拉组合框，它既具有列表框的功能，又具有文本框的功能，既可以从已给出的列表中选择数据项，也可以由用户直接输入数据项。

组合框的使用方式是通过 Style 属性设置的。Style 值为 0 时，组合框的形式为下拉组合框；Style 的值为 2 时，组合框的形式为下拉列表框，如图 5-36 所示。



图 5-35 组合框控件



图 5-36 组合框的分类

下拉列表框具有列表框和文本框的大部分属性，也有列表框的常用方法；下拉组合框可以输入内容，但只能通过 AddItem 方法和 RowSource 属性输入。组合框控件的常用属性如表 5-25 所示。



表 5-24 组合框控件的常用属性

| 属 性           | 功 能                                                                 | 默 认 值 |
|---------------|---------------------------------------------------------------------|-------|
| ControlSource | 指定与对象建立联系的数据源。一般需要先将在组合框中选择的数据项或输入的数据保存在相应的表中，再配置表单的数据环境            | 无     |
| DisplayCount  | 指定组合框中显示的数据项的个数                                                     | 0     |
| RowSource     | 指定组合框中数据项的来源，可以是表中字段，需要先设置 RowSourceType 属性方可使用                     | 无     |
| RowSourceType | 指定组合框中数据项来源的类型                                                      | 0-无   |
| Value         | 返回组合框中被选择的数据项。如果 Value 为数值型，表示被选择的数据项的序号；如果 Value 为字符型，表示被选择的数据项的内容 | 无     |

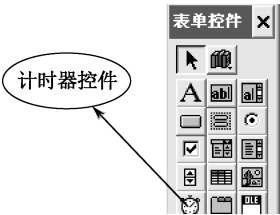


图 5-37 计时器控件

### 8. 计时器控件

计时器控件（Timer）用于创建计时器对象，如图 5-37 所示。计时器的作用是能够按照一定的时间间隔自动执行其 Timer 事件中的代码，可以用来处理反复执行的代码。

计时器是一个默认不可见的控件，无论将其放在表单的任何位置，运行表单时都不会显示计时器对象本身。

计时器控件最常用的事件为 Timer 事件，其常用的属性如表 5-26 所示。

表 5-25 计时器控件的常用属性

| 属 性 名    | 功 能                                                                                         | 默 认 值 |
|----------|---------------------------------------------------------------------------------------------|-------|
| Enabled  | 控制计时器控件是否可用                                                                                 | .T.   |
| Interval | 控制计时器 Timer 事件执行的时间间隔，值为 0 时，不触发事件，取值范围为 $0 \sim 2^{31}-1$ ，一般将值设置为 1000，单位为 ms(毫秒)，即每秒执行一次 | 0     |



## 实施步骤

### 1. 建立主表单

新建表单，命名为 frmManage，实现对 Student 表中数据进行处理并显示的功能。设置表单属性如表 5-27 所示。

表 5-26 frmManage 表单的属性

| 属 性 名        | 属 性 值    |
|--------------|----------|
| AlwaysOnTop  | .T.      |
| AutoCenter   | .T.      |
| Caption      | 学生成绩管理   |
| ShowWindow   | 1-在顶层表单中 |
| MoveAble     | .F.      |
| ClipControls | .F.      |
| MinButton    | .F.      |
| MaxButton    | .F.      |

2. 添加图像控件

单击控件工具箱中的“图像控件”按钮，在表单的适当位置拖曳出图像框，在其 Picture 属性中选择图像来源；将图像 Image1 的 Stretch 图像修改为“2-变比填充”。

3. 添加图形控件

使用图形控件 Shape 占位，分别将 Shape 的 SpecialEffect 设置为 3 维，效果如图 5-38 所示。

4. 添加命令按钮组控件

在右侧的 Shape 控件上拖曳出一个命令按钮组控件，将其 ButtonCount 属性值设置为 4，即组中包含 4 个命令按钮。右击该按钮组，在弹出的快捷菜单中选择“编辑”命令，再逐个单击其中的命令按钮，将 4 个命令按钮的 Caption 属性分别设置为“添加”、“修改”、“删除”和“关闭”。效果如图 5-28 所示。



图 5-38 使用 Shape 图形控件占位

5. 添加表格控件

在左侧的 Shape 控件上拖曳出一个表格控件 Grid1，将其 ColumnCount 属性值设置为 11（Student 表中共有 11 个字段）。右击该表格，在弹出的快捷菜单中选择“编辑”命令，选中 Column1 的 Header1，修改其 Caption 属性值为学号，依照 Student 表为表格其他列的 Header 对象修改 Caption 属性。

选择“显示”菜单中的数据环境，添加 Student 表，参照表 5-28 设置 Grid1 的属性。

表 5-27 Grid1 的属性

| 属 性 名            | 属 性 值   |
|------------------|---------|
| RecordSourceType | 1-别名    |
| RecordSource     | Student |
| ReadOnly         | .T.——真  |

单击工具栏中的“运行”按钮，查看表单运行效果，不断调试使其美观实用。效果如图 5-28 所示。

6. 在菜单中运行表单



图 5-39 在菜单中使用运行表单命令

修改“学生管理”菜单中的“学生信息维护”子菜单的结果为“命令”，使用户单击此菜单项时能够运行表单 frmManage.scx。命令为 do form frmManage，如图 5-39 所示。

7. 设计“学生信息维护”表单的操作流程

使用命令按钮组控件实现学生信息的添加、修改、删除与取消操作。



双击命令按钮组控件，在其 Click 事件中输入如下代码：

```
DO CASE
CASE This.Value =1 &&运行添加记录表单
 DO FORM frmAddStudent
CASE This.Value =2 &&运行修改记录表单
PUBLIC N
N=RECNO()
 DO FORM frmChangeStudent
CASE This.Value =3 &&物理删除表中已逻辑删除的记录
SET DELETED OFF
USE Student
Y=.F. &&设置删除标记 Y，初始值为假
 GO TOP
&&开始遍历表记录，遇见第一个逻辑删除记录时，修改 Y 值，并退出遍历
 DO WHILE NOT EOF()
IF Deleted()
 Y=.T.
 EXIT
ENDIF
SKIP
ENDDO
IF NOT Y &&表中没有已逻辑删除记录
 MessageBox(" 请先添加删除标记！ ")
ELSE
 RESULT=MESSAGEBOX(" 确定要彻底删除选中的记录吗？ ",4+32," 彻底删除 ")
 IF RESULT=6 &&消息对话框返回值为“确定”，执行物理删除命令
 PACK
 MessageBox (" 已彻底删除标记的记录！ ")
 ENDIF
ENDIF
ThisForm.Refresh
RELEASE Y
OTHERWISE
 ThisForm.Release
ENDCASE
```

## 8. 制作表单

制作“添加学生记录”表单，表单效果如图 5-28 所示。

新建表单 frmAddStudent.scx，设置表单的数据环境为表 Student.dbf，将 Student 表中所有字段拖动到表单中，快速绑定表单控件与表中字段，并按照图 5-28 排列各字段位置。

1) 将出生日期文本框更改为“出生日期”控件，效果如图 5-40 所示。

使用 Ole 对象控件显示并修改出生日期信息。单击控件工具箱中的“ActiveX(OleControl)”控件，选中“插入控件”单选按钮，在“控件类型”列表框中选择“Microsoft Date and Time Picker Control 6.0”选项，如图 5-41 所示，单击“确定”按钮即可插入该对象。

将其 Name 属性修改为“Ole 出生日期”。

2) 将“性别”文本框更改为“性别”选项按钮组控件。

显示性别信息，使用选项按钮组控件，添加控件并将 Name 属性修改为“opt 性别”，右击该控件，选择“编辑”命令，即可逐个编辑按钮组中的每个选项按钮。



分别将两个选项按钮的 Caption 属性值修改为“男”和“女”。



图 5-40 日期控件效果

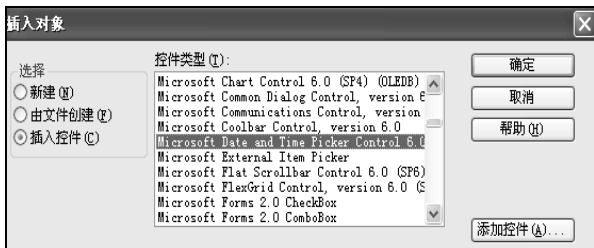


图 5-41 选择日期控件

### 3) 添加“照片”控件。

增加字段“照片路径”，类型为备注型（二进制），如图 5-42 所示。

显示照片使用图像控件，控件名为“img 照片”。

在图像下方添加文本框控件，控件名为“txt 照片地址”。

在文本框右侧添加命令按钮，名为“cmd 浏览”，其 Caption 属性值为“浏览”，作用是打开“选择图像”对话框并选择图像。

在“cmd 浏览”按钮的 Click 事件输入如下代码：

```
ThisForm.txt 照片地址.Value =GetPict()
ThisForm.img 照片.Picture = ThisForm.txt 照片地址.Value
```

其中，系统函数 GetPict()的作用是显示“打开图像”对话框，并返回选定图片文件的文件名。

4) 添加“专业”控件。添加组合框控件，并将控件的 Name 属性修改为“comb 专业”，设置其 RowSourceType 属性为“1-值”，RowSource 属性为“计算机网络技术，电子技术，电子商务，动漫与游戏设计”。效果如图 5-43 所示。

### 5) 添加按钮组控件。

在所有控件下方添加命令按钮组控件，其 Name 属性为“cmd 操作”，将按钮组的 ButtonCount 属性值设置为 3。

将按钮组的 3 个命令按钮的 Caption 属性分别设置为“保存”、“重填”和“取消”。效果如图 5-44 所示。



图 5-42 备注型（二进制）的“照片路径”字段

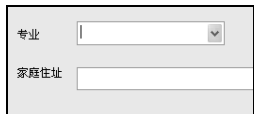


图 5-43 专业字段对应的下拉列表控件

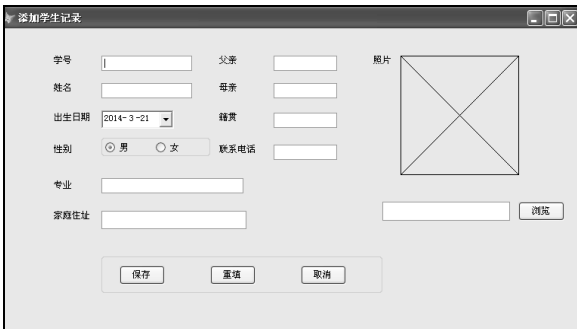


图 5-44 表单 frmAddStudent 整体效果图



在命令按钮组的 Click 事件中输入如下代码：

```
DO CASE
CASE This.Value=1
APPEND BLANK &&以下记录内容添加到新增的空记录中
REPLACE 父亲 WITH AllTrim(ThisForm.txt 父亲.Value)
REPLACE 籍贯 WITH AllTrim(ThisForm.txt 籍贯.Value)
REPLACE 联系电话 WITH AllTrim(ThisForm.txt 联系电话.Value)
REPLACE 母亲 WITH AllTrim(ThisForm.txt 母亲.Value)
REPLACE 姓名 WITH AllTrim(ThisForm.txt 姓名.Value)
REPLACE 学号 WITH AllTrim(ThisForm.txt 学号.Value)
REPLACE 专业 WITH AllTrim(ThisForm.comb 专业.Value)
REPLACE 出生日期 WITH ThisForm.ole 出生日期._Value
REPLACE 性别 WITH IIF(ThisForm.opt 性别.Value=1, "男", "女")
REPLACE 家庭住址 WITH AllTrim(ThisForm.edt 家庭住址.Value)
APPEND MEMO 照片地址 FROM AllTrim(ThisForm.txt 照片地址.Value)
ThisForm.img 照片.PictureVal=照片地址
MessageBox(" 信息添加成功！请单击右上角的关闭按钮！")
CASE This.Value =2
ThisForm.txt 父亲.Value =Space(0)
ThisForm.txt 籍贯.Value = Space(0)
ThisForm.txt 联系电话 .Value = Space(0)
ThisForm.txt 母亲.Value = Space(0)
ThisForm.txt 姓名.Value = Space(0)
ThisForm.txt 学号.Value = Space(0)
ThisForm.txt 照片地址.Value = Space(0)
ThisForm.comb 专业.Value = Space(0)
ThisForm.ole 出生日期._Value =Date()
ThisForm.opt 性别.Value =0
ThisForm.edt 家庭住址.Value =Space(0)
ThisForm.img 照片.Picture=Space(0)
CASE This.Value=3
GO BOTTOM
DELETE
PACK
ThisForm.Release
ENDCASE
```

## 9. 制作“修改学生信息”表单

创建表单文件 frmChangeStudent.scx ,该表单的 Name 属性为 frmChangeStudent ,Caption 属性值为“修改学生信息”，其他属性参考“添加学生记录”表单的各属性。

图 5-45 “修改学生信息”表单

参照“添加学生记录”表单界面，在“修改学生信息”表单中添加控件，如图 5-45 所示。

在表单的 Init 事件中输入如下代码：

```
IF Used(" Student ")
SELECT Student
ELSE
USE Student IN 0
ENDIF
GO N
```

&&N 是在父表单 frmManage 中定义的公共变量，保存当前记录号

```
ThisForm.Comb 专业.Value =专业
ThisForm.Img 照片.PictureVal=照片地址
```

在命令按钮组的 Click 事件中输入如下代码：

```
DO CASE
CASE This.Value=1
 &&以下内容的修改针对当前第 N 条记录
 REPLACE 父亲 WITH AllTrim(ThisForm.txt 父亲.Value)
 REPLACE 籍贯 WITH AllTrim(ThisForm.txt 籍贯.Value)
 REPLACE 联系电话 WITH AllTrim(ThisForm.txt 联系电话.Value)
 REPLACE 母亲 WITH AllTrim(ThisForm.txt 母亲.Value)
 REPLACE 姓名 WITH AllTrim(ThisForm.txt 姓名.Value)
 REPLACE 学号 WITH AllTrim(ThisForm.txt 学号.Value)
 REPLACE 专业 WITH AllTrim(ThisForm.comb 专业.Value)
 REPLACE 出生日期 WITH ThisForm.ole 出生日期._Value
 REPLACE 性别 WITH IIF(ThisForm.opt 性别.Value=1, "男", "女")
 REPLACE 家庭住址 WITH AllTrim(ThisForm.edt 家庭住址.Value)
 APPEND MEMO 照片地址 FROM AllTrim(ThisForm.txt 照片地址.Value)
 ThisForm.img 照片.PictureVal=照片地址
 MESSAGEBOX(" 信息修改成功！请单击右上角的关闭按钮！")
CASE This.Value =2
 ThisForm.txt 父亲.Value =Space(0)
 ThisForm.txt 籍贯 .Value = Space(0)
 ThisForm.txt 联系电话 .Value = Space(0)
 ThisForm.txt 母亲 .Value = Space(0)
 ThisForm.txt 姓名 .Value = Space(0)
 ThisForm.txt 学号 .Value = Space(0)
 ThisForm.txt 照片地址.Value = Space(0)
 ThisForm.comb 专业.Value = Space(0)
 ThisForm.ole 出生日期._Value =Date()
 ThisForm.opt 性别 .Value =0
 ThisForm.edt 家庭住址 .Value =Space(0)
 ThisForm.img 照片.Picture=Space(0)
CASE this.Value =3
 ThisForm.Release
ENDCASE
```

## 任务二 实现“学生信息查询”功能

### 任务目标

能制作“学生管理”菜单下的“学生信息查询”子菜单项对应的功能。

具体包括：设置查询条件；显示查询结果；关闭“查询”窗口。

### 任务描述

使用页框等控件设计并实现“学生信息查询”表单功能，效果如图 5-46 所示。

图 5-46 查询设置表单



## 相关知识

### 1. 页框控件

页框控件（PageFrame）用于创建页框对象，如图 5-47 所示。页框的作用是将具有一定关系的控件组织在一起，既可以美化应用程序的界面，又可以节省表单空间。页框包括多个页面，每个页面又称为选项卡。例如，Word 2003 中的“查询和替换”对话框中包括“查找”、“替换”和“定位”3 个页面，每个页面都可以看成一个表单，在其中可以添加其他控件，各个控件之间是相互独立的，如图 5-48 所示。



图 5-47 页框控件



图 5-48 “查找和替换”对话框

页框控件、页框中的每个页面和页面中的所有控件都具有各自的属性、事件和方法。一个页框控件可以包括多个页面，但是在多个页面中只能有一个活动页面，就像打开的多个 Windows 窗口中只能有一个为活动窗口一样。页框控件的常用属性如表 5-29 所示。

表 5-28 页框控件的常用属性

| 属 性            | 功 能          | 默 认 值 |
|----------------|--------------|-------|
| PageCount      | 指定页框中页面的个数   | 2     |
| TabOrientation | 指定选项卡在页框中的位置 | 0 (上) |

页框中的页面具有属于自己的属性，页面常用的属性如表 5-30。

表 5-29 页面的常用属性

| 属 性       | 功 能       | 默 认 值        |
|-----------|-----------|--------------|
| Caption   | 指定页面的标题   | Page1, Page2 |
| PageOrder | 指定页面的相对顺序 | 1            |

说明：在页面中添加控件时，需要先在页框上右击，在弹出的快捷菜单中选择“编辑”命令，当页框的边框变为浅绿色编辑状态时，再选中活动页面，即可添加控件，否则会将控件添加在表单上。



## 实施步骤

“学生信息查询”表单用于查询学生信息，可以通过学号、姓名、专业、籍贯和联系电话查找学生信息，效果如图 5-46 所示。

### 1. 创建表单

使用表单设计器新建表单，保存为 frmSearch.scx，参照表 5-31 修改表单的属性；设置表

单的数据环境为表 Student。

表 5-30 查询表单的常用属性

| 属 性 名       | 属 性 值     |
|-------------|-----------|
| AutoCenter  | .T.       |
| Caption     | 学生信息查询    |
| Name        | frmSearch |
| ControlBox  | .F.       |
| ShowWindows | 1-在顶层表单中  |

2. 添加页框控件

在控件工具箱中选择页框控件，在表单的适当位置拖曳鼠标即可，设置页框控件的 PageCount 属性值为 2，右击页框，在弹出的快捷菜单中选择“编辑”命令，设置两个页框的 Caption 属性值分别为“查询设置”和“查询结果”。

3. 在“查询设置”页框中添加控件

在“查询设置”页框中添加矩形控件，用于显示查询条件背景。右击页框控件，在弹出的快捷菜单中选择“编辑”命令，即可在页框中添加控件或进行修改。

在矩形控件上添加标签和文本框，包括“学号”、“姓名”、“专业”、“籍贯”和“联系电话”，并调整标签控件和文本框控件的大小和对齐方式。

4. 在“查询结果”页框中添加控件

右击编辑页框，在页框 2 中添加表格控件，调整表格大小，如图 5-49 所示。

5. 设计“并/或”关系

使用标签控件显示“各项之间关系”文本，使用选项按钮组控制多个条件之间的关系。

在控件工具箱中选择“标签”控件，将其 Caption 属性值设置为“各项之间关系”，并设置其 AutoSize 属性值为“.T.”。

在控件工具箱中选择“选项按钮组”控件，设置其 ButtonCount 属性值为 2，Value 属性值设置为 1。右击按钮组，在弹出的快捷菜单中选择“编辑”命令，分别将两个选项按钮的 Caption 属性值修改为“并且”和“或者”。

6. 设计命令按钮

添加“查询”按钮，设置其 Caption 属性为“查询”，Default 属性值为“.T.”。双击打开“查询”按钮的 Click 过程，在其中输入如下代码：

```
&&判断多个条件间是“并且”还是“或者”的关系
IF ThisForm.PageFrame1.Page1.OptionGroup1.Value =1 THEN
 ANDOR= " AND "
```

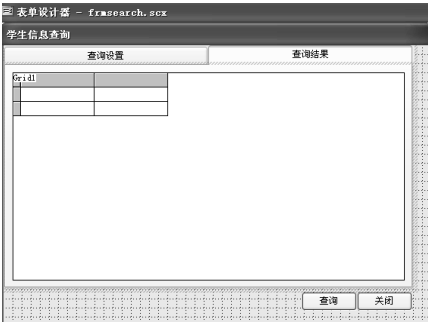


图 5-49 查询结果表单



```
ELSE
 ANDOR= " OR "
ENDIF
&&初始化查询连接字符串
whereCon= " "
&&判断如果学生姓名条件不为空，则以姓名为模糊查询的条件
IF NOT Empty(AllTrim(ThisForm.PageFrame1.Page1.txtStudentName .Text)) THEN
 whereCon= " (姓名 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtStudentName .Text)+ "%') "
ENDIF
&&如果学号不为空，查询条件包括姓名和学号
IF NOT Empty(AllTrim(ThisForm.PageFrame1.Page1.txtStudentNo.Text)) THEN
 IF Empty(whereCon) THEN
 whereCon= " (学号 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtStudentNo.Text)+ "%') "
 ELSE
 whereCon=whereCon+ANDOR+ " (学号 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtStudentNo.Text)+ "%') "
 ENDIF
ENDIF
&&在已有的姓名、学号作为查询条件的基础上，增加专业作为查询条件
IF NOT Empty(AllTrim(ThisForm.PageFrame1.Page1.txtDepartment .Text)) THEN
 IF Empty (whereCon) THEN
 whereCon= " (专业 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtDepartment.Text)+ "%') "
 ELSE
 whereCon=whereCon+ANDOR+ " (专业 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtDepartment .Text)+ "%') "
 ENDIF
ENDIF
&&在已有的姓名、学号、专业作为查询条件的基础上，增加籍贯作为查询条件
IF NOT Empty(AllTrim(ThisForm.PageFrame1.Page1.txtOrigo.Text)) THEN
 IF Empty (whereCon) THEN
 whereCon= " (籍贯 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtOrigo .Text)+ "%') "
 ELSE
 whereCon=whereCon+andor+ " (籍贯 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtOrigo .Text)+ "%') "
 ENDIF
ENDIF
&&增加联系电话作为查询条件
IF NOT Empty(AllTrim(ThisForm.PageFrame1.Page1.txtTel.Text)) THEN
 IF Empty (whereCon) THEN
 whereCon= " (联系电话 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtTel.Text)+ "%') "
 ELSE
 whereCon=whereCon+andor+ " (联系电话 Like '%" +;
 AllTrim(ThisForm.PageFrame1.Page1.txtTel.Text)+ "%') "
 ENDIF
ENDIF
&&指定数据环境并设置查询语句 ExecSqlStr
IF Used(" Student ")
 SELECT " Student "
ELSE
 SELECT 0
```

```

 USE Student
 ENDIF
 IF NOT Empty(whereCon) THEN
 ExecSqlStr= " SELECT * FROM Student Where " +whereCon+ " INTO CURSOR
NewTable "
 ELSE
 ExecSqlStr = "SELECT * FROM Student "+ " INTO CURSOR NewTable "
 ENDIF
 &&设置表格及页框的属性，使其执行查询
 ThisForm.PageFrame1.Page2.Grid1.RecordSourceType = 4
 ThisForm.PageFrame1.Page2.Grid1.RecordSource = ExecSqlStr
 ThisForm.PageFrame1.ActivePage =2
 ThisForm.PageFrame1.Page2.Grid1.SetFocus
 ThisForm.PageFrame1.Page2.Refresh

```

添加“关闭”按钮，设置其 Caption 属性值为“关闭”，并在其 Click 事件中输入如下代码：

```

CLOSE TABLES ALL
ThisForm.Release

```



## 问题探究

### 1. 如何读取命令按钮组或选项按钮组中的某个具体对象？

命令按钮组或选项按钮组这样的容器类控件可以作为一个整体来使用，也可以直接使用其上的每个组成控件。下面以命令按钮组为例说明使用方法。

例如，要通过一个命令按钮组实现显示表格中的每一条记录的功能。

建立表单，Caption 属性设置为“显示表中任意记录”。在表单的数据环境中添加 Student 表，并拖动其各个控件到表单中，使它们与表相关联。

添加命令按钮组，设置其 ButtonCount 值为 4，即包含 4 个命令按钮。右击命令按钮组，在弹出的快捷菜单中选择“编辑”命令，分别将每个命令按钮的 Caption 属性设置为“第一条”、“上一条”、“下一条”、“最后一条”。

输入功能代码。

方法一：在命令按钮组的 Click 事件中输入如下代码：

```

DO CASE
CASE This.Value=1
GO TOP
CASE This.Value=2
SKIP -1
CASE This.Value=3
SKIP
OTHERWISE
GO BOTTOM
ENDCASE
IF EOF()
MessageBox(“已是最后一条记录”)
GO BOTTOM
ENDIF
ThisForm.Refresh

```

方法二：右击命令按钮组，在弹出的快捷菜单中选择“编辑”命令，再分别选中每一个命令按钮，在它们的 Click 事件中分别输入对应的代码。



“第一条”按钮的代码如下：

```
GO TOP
ThisForm.Refresh
```

“下一条”按钮的代码如下：

```
SKIP
IF EOF()
 MessageBox(“已是最后一条记录”)
GO BOTTOM
ENDIF
ThisForm.Refresh
```

“上一条”按钮的代码如下：

```
SKIP -1
ThisForm.Refresh fresh
```

“最后一条”按钮的代码如下：

```
GO BOTTOM
ThisForm.Refresh
```

## 2. 如何使用计时器实现计数功能？

在项目管理器中建立表单“计数器.scx”，选择“文件” “保存”命令。

在打开的表单设计中添加两个标签控件，两个文本框控件和两个命令按钮控件，属性如表 5-32 所示。

表 5-31 各个控件的属性

| 控 件      | 属 性 名     | 属 性 值 |
|----------|-----------|-------|
| Label1   | Caption   | 系统时间  |
|          | AutoSize  | .T.   |
| Label2   | Caption   | 计数器   |
| Text1    | AutoSize  | .T.   |
|          | ReadOnly  | .T.   |
|          | Alignment | 1-右   |
| Text2    | ReadOnly  | .T.   |
|          | Alignment | 1-右   |
| Command1 | Caption   | 开始计数  |
| Command2 | Caption   | 停止计数  |

调整各个控件的位置（可以使用“布局”工具栏），效果如图 5-50 所示。

在控件工具箱中单击选中计数器控件，将其拖动到表单中，可以放在任意位置。在右侧的属性面板中，将其 Interval 属性值设置为 1000，如图 5-51 所示。



图 5-50 表单设计器

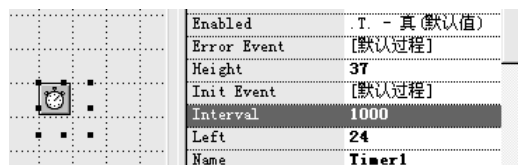


图 5-51 计时器的 Interval 属性



设置表单的初始化资源。在表单的空白位置双击，打开表单的代码窗口，如图 5-52 所示。

在表单的 Init 事件中输入如下代码：

```
ThisForm.Text1.Value=Time()
ThisForm.Text2.Value=0
```

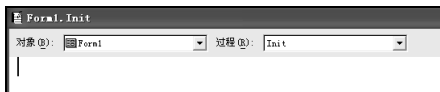


图 5-52 表单初始化事件的代码窗口

双击计时器控件 Timer1 ( 表单中第一个计时器控件的名称 )，在其 Timer 事件中输入如下代码：

```
ThisForm.Text1.Value=Time() &&显示系统时间
ThisForm.Text2.Value= ThisForm.Text2.Value+1 &&使用计时器计时
```

在“开始计数”命令按钮的 Click 事件中输入如下代码：

```
ThisForm.Timer1.Enabled=.T.
```

在“停止计数”命令按钮的 Click 事件中输入如下代码：

```
ThisForm.Timer1.Enabled=.F.
```

单击工具栏中的“保存”按钮，运行表单。



## 阅读材料

### 1. 控件的分类

Visual FoxPro 系统中最常用的表单控件大致可以分为以下几类。

输出类控件：标签、线条、形状、图像、分隔线。

输入类控件：文本框、编辑框、列表框、组合框、微调控件。

容器类控件：命令按钮组、选项按钮组、表格、页框、容器。

连接类控件：ActiveX 控件、ActiveX 绑定控件、超链接。

### 2. 向数据表插入图像控件中显示的图像

在数据表中插入图像或在表单中显示数据表中存储的图像都可以使用 Image 控件来处理。Visual FoxPro 9.0 是使用 GDI ( Graphics Device Interface ) 来输出图形的，它支持 BMP、JPEG、PNG、TIFF 等图像格式。直接将图像嵌入到数据库中来显示图像并不是速度最快的方法，应将图像存放在磁盘上，在数据库中使用备注型 ( 二进制 ) 类型的字段存放文件名。当要修改已存放在数据表中的图像时，使用命令：REPLACE 字段名 WITH FileToStr ( 图像文件名 )；要显示图像时，图像控件 Image1.PictureVal = 字段名。

### 3. 表格控件的组成

表格控件是一种容器类对象，一个表格由若干个列对象组成，这个原理与命令按钮组、选项按钮组等容器类控件都是一致的，它们都是由若干个对象组成的。但是，由于列本身也是一个容器类对象，它可以包含若干个列标题和其他控件，所以表格是一个嵌套容器类对象。表格、列、列标题及其他控件都有各自的属性、事件和方法，使用时要注意它们之间的层次关系。

### 4. 一对多表格

表格中最常见的一种应用就是父子表格，也称为一对多表格，对应一对多联系。当用户选中父表格中的某个记录时，子表格中就会显示出与该记录相关联的信息。要创建一对多表格，则在父子表格中显示的两个数据表应当建立表间的一对多关系。



表间的一对多关系可以直接在数据环境中进行,也可以预先在数据库设计器中完成(这在前面单元中已经介绍)。例如,创建一个显示学生信息及其成绩信息的一对多表格,步骤如下。

创建一个表单,设置其 Caption 属性为“学生及成绩信息”。将 Student 和 SelCourse 两个表添加到表单的数据环境中,依次将两个表从数据环境中拖放到表单中,调整其大小位置。

如果 Student 和 SelCourse 两个表已经存在关联,则不需要再建立关系,否则要根据学号建立两个表的一对多关联。例如,使用生成器创建表间关联。右击子表格 SelCourse 控件,在弹出的快捷菜单中选择“生成器”命令,在“关联”选项卡中分别选择父表的关键字段和子表的关键字段,这里都是学号字段。

保存表单,调试并运行表单,观察效果。



## 项目小结

本项目是在项目三的基础上继续完成“学生成绩管理信息系统”的各项功能。项目三通过对系统菜单的设计,显示系统的总体功能,具体要将这些功能都实现,就需要应用表单及其控件来完成。本项目通过完成“学生成绩管理信息系统”的各个菜单项,详细介绍表单中各个控件的使用方法,包括标签、文本框、编辑框、列表框、组合框、命令按钮、命令按钮组、选项按钮组、页框、表格、计时器、图像等控件的常用属性、事件和方法。读者可以通过本项目完成其他类似系统的功能模块的设计。



## 上机实习

1. 参照“学生成绩管理信息系统”的“学生管理”子菜单项的表单设计,制作“课程管理”子菜单项的表单设计。

制作课程信息维护表单。

制作课程信息查询表单。

2. 参照“学生成绩管理信息系统”的主表单设计,制作“图书管理信息系统”的主表单设计。

制作图书维护表单,包括图书的添加、修改、统计与关闭。

制作图书查询表单,如图 5-53 所示。

图 5-53 “图书管理信息系统”的“图书查询”表单



## 知识巩固

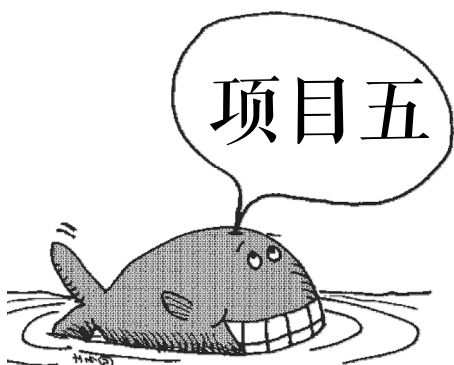
### 一、选择题

1. 表格控件需要先设置 ( ) 属性,再设置 ( ) 属性,才可以使其与数据源绑定。  
A. RowSourceType                      B. RowSource  
C. RecordSourceType                    D. RecordSource
2. 命令按钮组的 ( ) 属性可以确定选择的是组中的哪个命令按钮。  
A. Value                      B. Button                      C. Caption                      D. ButtonCount
3. 一个选项按钮组中可以同时选中 ( )。

- A. 一个                  B. 两个                  C. 3 个                  D. 任意个
4. 编辑框一般用于显示 ( ) 的数据。  
A. 备注型                  B. 日期型                  C. 数据类型                  D. 通用型
5. 下列选项中, ( ) 控件既能实现列表框的功能, 同时又节省表单空间。  
A. 文本框                  B. 编辑框                  C. 页框                  D. 组合框
6. 下列控件中, ( ) 是运行时自动隐藏的。  
A. 标签                  B. 文本框                  C. 命令按钮                  D. 计时器
7. 要在指定的时间间隔下, 不断执行相应的代码, 应将代码写在计时器控件的 ( ) 事件中。  
A. Click                  B. DblClick                  C. Time                  D. Timer
8. 很多表单控件都具有 Caption 属性, 无论什么控件的 Caption 属性, 都只能接收 ( ) 的数据。  
A. 字符型                  B. 数值型                  C. 日期型                  D. 任意类型

## 二、填空题

1. 表格控件的\_\_\_\_\_属性表示表格中显示的列的个数。
2. 命令按钮组是\_\_\_\_\_类控件。
3. 计时器的\_\_\_\_\_属性可以确定计时器的时间间隔。
4. Visual FoxPro 的控件分为\_\_\_\_\_、\_\_\_\_\_输出类和输入类 4 种。
5. 在 Label1 中显示当前系统时间的命令是\_\_\_\_\_。



## 创建学生信息报表

### 项目说明

在 Visual FoxPro 的应用系统中,经常需要将数据处理的结果以报表的形式打印出来或显示在屏幕上。报表是输出数据处理结果的一种常用方式,对应用程序来说,报表是很重要的一个组成部分。报表的使用很灵活,它可以将结果输出到屏幕窗口进行浏览显示,也可以打印到打印机上,或可以输出到用户指定的其他文件中。

在本项目中为添加或修改过的学生信息、课程信息、成绩信息制作报表,是为了将操作的结果以打印的形式呈现给用户。当使用者需要根据统计结果进行决策或分析时,打印的效果往往是最好的。设计优秀的打印报表是 Visual FoxPro 设计者最终成果的体现。

本项目实施过程:

任务:设计并制作“学生管理”中的“学生信息报表”。

### 任务 设计并制作“学生管理”中的“学生信息报表”

#### 任务目标

能制作学生信息报表。

能够对报表中的数据进行分组计算。

#### 任务描述

创建学生信息报表,报表打印预览效果如图 5-54 所示。

| 专业   | 学号     | 姓名  | 出生日期     | 性别 | 籍贯  | 父亲 | 母亲 | 联系电话 | 家庭住址 |
|------|--------|-----|----------|----|-----|----|----|------|------|
| 电子技术 | 120201 | 金明  | 02/09/99 | 男  | 吉林  |    |    |      |      |
|      | 120202 | 唐莎  | 10/10/98 | 女  | 吉林  |    |    |      |      |
|      | 120203 | 王敏敏 | 01/20/99 | 女  | 黑龙江 |    |    |      |      |
|      | 120204 | 李璐  | 07/08/98 | 女  | 山东  |    |    |      |      |

图 5-54 学生信息报表效果图

## 相关知识

报表是一种类型的文件，其扩展名为“.frx”。报表包括两个组成部分：数据源和布局。数据源是指要打印的数据表或视图；布局指报表的打印格式。

### 1. 使用报表向导创建报表

Visual FoxPro 提供了报表设计向导，用户可以根据向导给出的提示步骤设置报表的组成。报表向导分为简单报表和一对多报表。简单报表指报表的数据源为一个表；一对多报表指数据源包含了父表的记录和相关子表的记录，如图 5-55 所示。

### 2. 使用报表设计器创建报表

使用报表创建向导可以方便、快捷地将数据源打印输出，但报表的布局是系统事先定义好的，如果用户想要设计更加美观和个性化的报表布局，可以使用报表设计器来完成。使用报表设计器创建报表，就像创建表单一样，选择“报表控件”工具栏中的报表控件，将其添加在报表设计器中即可，如图 5-56 所示。



图 5-55 报表向导



图 5-56 报表设计器及“报表控件”工具栏

通常使用报表向导创建基础报表，再使用报表设计器进行详细的修改。报表设计器中默认包括页标头、细节和页注脚 3 个带区。

页标头：用于列出每页标题只显示一次的数据或信息。

细节：用于显示每条记录的具体内容。



页注脚：用于列出每页最后只显示一次的数据或信息。

除了上述 3 个默认带区外，报表还可以包括标题和总结带区。如果为报表创建分组，还会增加组标题和组注脚。

选择报表的带区可以在报表设计器中右击，在弹出的快捷菜单中选择“可选带区”命令即可。

### 3. 分组报表

使用报表进行打印输出时，通常需要将具有某种相同特征的数据组织在一起输出，使得报表界面更加美观和整洁，这时需要使用报表的分组功能。



图 5-57 “报表设计器”中的数据分组控件

使用报表向导创建报表时，直接设置分组字段即可；使用报表设计器分组时，可以使用报表设计器工具栏中的“数据分组”控件，通过分组字段控件的表达式设置分组，如图 5-57 所示。

### 4. 预览和打印报表

#### (1) 预览报表

当用户将报表设计完毕时，可以使用系统提供的报表预览功能在屏幕上预览报表的设计效果，预览结果与实际打印结果完全一致。

在报表设计器打开的状态下，选择“显示”“预览”命令，即可观察预览效果。

在应用程序中可以使用命令预览报表，命令如下：

```
REPORT FORM 报表文件名 PREVIEW
```

#### (2) 打印报表

如果已进行打印设置，在报表设计器的状态下，选择“文件”“打印”命令，即可打印报表。

在应用程序中可以使用命令打印报表，命令如下：

```
REPORT FORM 报表文件名 TO PRINTER
```

### 5. 标签

标签是一种简易的报表，报表的数据源为一个表，而标签的数据源为表中的一条记录。例如，为员工打印工资条时，就可以使用标签。标签可以为指定的纸张设计打印效果，是一种特殊的报表。标签文件的扩展名为“.lbx”。使用标签创建向导，可以快速地创建标签文件，在向导中，系统提供了创建标签文件的步骤，用户可以根据提示信息完成标签的创建。

应用程序中可以使用命令对标签文件进行预览，预览标签的命令如下：

```
LABEL FORM 标签文件名 PREVIEW
```

在应用程序中可以使用命令将标签文件打印出来，打印标签的命令如下：

```
LABEL FORM 标签文件名 TO PRINTER
```



## 实施步骤

#### 1. 建立报表文件

使用报表向导创建报表。

1) 选择报表数据源中的字段，如图 5-58 所示。

2) 选择分组记录, 如图 5-59 所示。



图 5-58 字段选取

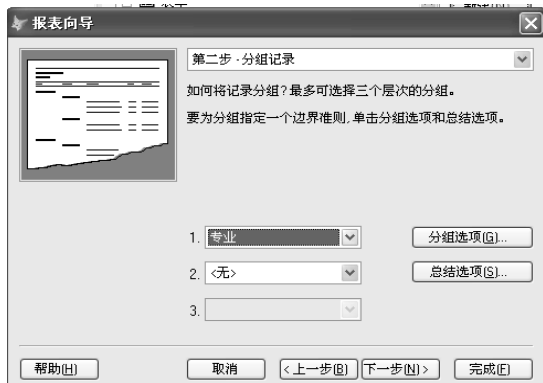


图 5-59 分组记录

3) 选择报表样式, 如图 5-60 所示。

4) 定义报表布局, 如图 5-61 所示。

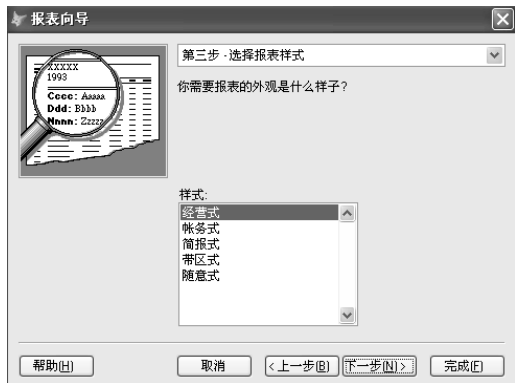


图 5-60 选择报表样式

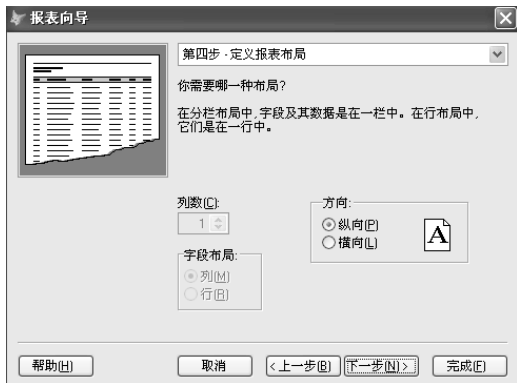


图 5-61 定义报表布局

5) 选择排序字段及排序方式, 如图 5-62 所示。

6) 完成报表向导, 可选择预览按钮观察效果, 如图 5-63 所示。

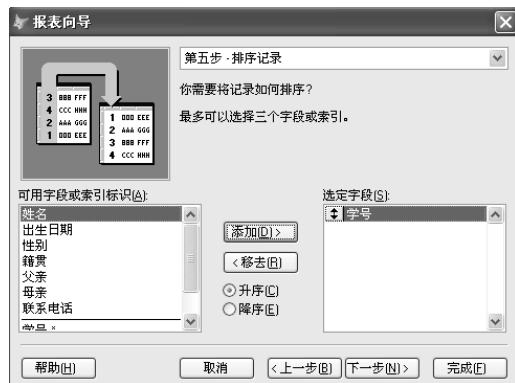


图 5-62 排序记录



图 5-63 完成报表向导

7) 保存名为 frxStudent.frx, 预览报表效果, 如图 5-64 所示。

8) 选中报表, 使用报表设计器对报表进行修改。改变报表中列标题 (即字段名) 对应的



标签位置和大小,使报表美观。报表中字段内容是由域控件显示的。

由于报表中字段较多,将报表页面设置为横向。选择“报表”“属性”命令,弹出“报表属性”对话框,在“页面布局”选项卡中选择“页面设置”中的“横向”选项,如图 5-65 所示。



图 5-64 预览报表



图 5-65 报表的页面设置

可以删除某些列,如照片地址等,调整列标题位置及细节中的文本框位置,并调整布局中的直线长度,使效果美观,如图 5-66 所示。

9) 使用布局工具箱调整多个控件的对齐方式,如图 5-67 所示。选中多个控件,单击工具箱中的第三个按钮,使它们靠上对齐。

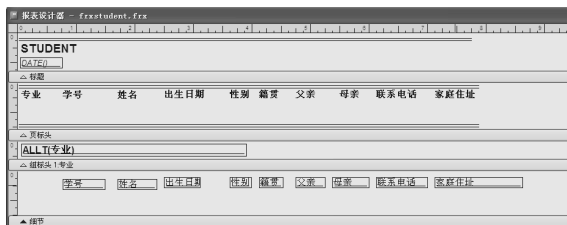


图 5-66 调整报表中控件的位置

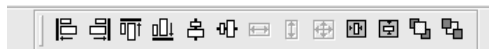


图 5-67 报表的布局工具箱

最终制作出的报表预览效果如图 5-54 所示。

## 2. 在菜单中调用报表

在 Caidan.mnx 的菜单设计器中打开“学生信息维护”子菜单,在“结果”选项组中选择“命令”选项,输入如下命令:REPORT FORM frxStudent PREVIEW 如图 5-68 所示。重新生成菜单,运行并观察效果。



## 问题探究



图 5-68 在菜单中应用报表

### 1. 如何确定报表设计器与报表向导的使用场合?

报表向导是建立报表最快速的方式,创建报表的界面也比较美观,但使用报表向导创建时灵活性不够,需要使用报表设计器对其进一步调整。可以先用报表向导创建报表框架,再应用报表设计器微调。

当报表的数据源包含多个表时,使用一对多报表向导



导更方便。



## 阅读材料

### 1. 报表的保护

Visual FoxPro 9.0 为报表设置了保护选项,用户可以选取在保护模式下报表版面的一些不可修改的特性如报表不能运行或打印,报表变量不能被编辑,选择的区带不能编辑等。用户可以通过选择“报表”“属性”命令,在弹出的“报表属性”对话框中设置“保护”选项卡,如图 5-69 所示。

### 2. 加载报表的数据环境

与表单相似,报表也有数据环境,利用数据环境可以方便地为报表添加数据源。Visual FoxPro 9.0 中为报表增加了一项“加载数据环境”功能,用户不用逐一添加数据环境,而是将已有报表的数据环境加载到当前报表中。

在“报表属性”对话框中选择“数据环境”选项卡,选中“从另一个报表文件复制”单选按钮,单击“选择”按钮,在弹出的“打开”对话框中选择一个已建报表,即可将该报表的数据环境复制到新报表中,如图 5-70 所示。

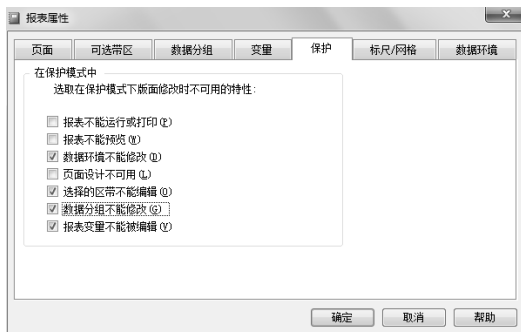


图 5-69 设置“报表属性”对话框中的“保护”选项卡



图 5-70 设置“报表属性”对话框中的“数据环境”选项卡



## 项目小结

本项目介绍了将表中数据处理的结果以报表的形式输出的方法,并简要介绍了标签的使用方法。读者可以以“学生信息报表”为参考,设计其他信息的报表。



## 上机实习

1. 参照“学生信息报表”制作“课程信息报表”。
2. 参照“学生信息报表”制作“成绩信息报表”。
3. 参照“学生信息报表”制作“图书管理信息系统”的报表,包括图书信息报表、读者信息报表、借阅信息报表。



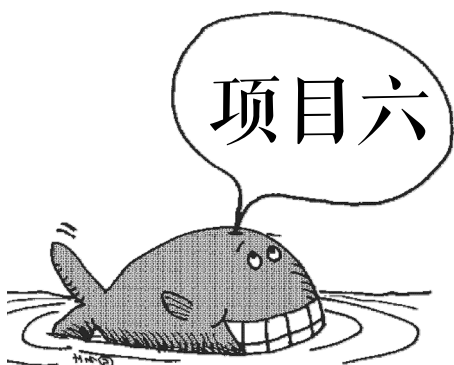
## 知识巩固

### 一、选择题

1. 报表文件的扩展名为 ( )。  
A. .mnx                      B. .frx                      C. .lbx                      D. .scx
2. 标签文件的扩展名为 ( )。  
A. .mnx                      B. .frx                      C. .lbx                      D. .scx
3. 若报表的数据源包含了父表的记录和相关子表的记录, 则应使用 ( )。  
A. 简单报表向导                      B. 一对一报表向导  
C. 一对多报表向导                      D. 多对多报表向导
4. 打印标签的命令是 ( )。  
A. REPORT FORM 报表文件名 PREVIEW  
B. REPORT FORM 报表文件名 TO PRINTER  
C. LABEL FORM 标签文件名 PREVIEW  
D. LABEL FORM 标签文件名 TO PRINTER
5. 报表中的字段内容通常使用 ( ) 控件显示。  
A. 标签                      B. 文本框                      C. 域                      D. 表格

### 二、填空题

1. 报表主要包括两个部分: \_\_\_\_\_ 和 \_\_\_\_\_。
2. 预览报表的命令是 \_\_\_\_\_。
3. 报表设计器包含页标头、页注脚和 \_\_\_\_\_, 当对报表中数据进行分组时, 还会出现 \_\_\_\_\_ 和 \_\_\_\_\_。



## 发布“学生成绩管理信息系统”

### 项目说明

应用程序开发完成后,需要将其制作成安装包,以便其他用户将其安装到自己的计算机上,这一过程称为程序的发布。Visual FoxPro 的程序发布是通过项目管理器中的“连编”来实现的。发布后的应用程序可以是可执行文件,不需要安装 Visual FoxPro 系统软件就可以执行,这为程序的移植提供了更大的空间。

本项目的实施过程如下:

任务一:连编“学生成绩管理信息系统”。

任务二:发布“学生成绩管理信息系统”。

### 任务一 连编“学生成绩管理信息系统”

#### 任务目标

能够连编“学生成绩管理信息系统”等应用程序。

#### 任务描述

将“学生成绩管理信息系统”项目文件连编成可执行文件,使其能够在其他机器上运行。

#### 相关知识

连编是指将项目管理器中的所有文件编译在一起,形成一个独立的可执行文件。连编后的可执行文件可以脱离 Visual FoxPro 系统环境独立运行,有利于应用系统的移植。连编时需要注意的是,要连编的所有文件必须都在一个项目管理器中,“学生成绩管理信息系统”的所有文件都是通过项目管理器建立的,因此这里可以直接进行连编。



## 1. 主程序

在连编前需要为整个应用系统设计一个主程序，即入口程序，它是应用程序最先运行的文件，它会调用应用程序的其他组件。被设置成主文件的文件名称会被加粗。

例如，在项目管理器中设置程序 main.prg 为主程序文件，并在主程序 main 中运行名为 frmLogin 的表单，则在 main.prg 程序中输入如下命令：

```
CLOSE ALL
_VFP.Visible=.F.
DO FORM frmLogin
READ EVENTS
CLEAR ALL
```

右击 main.prg 文件按钮，在弹出的快捷菜单中选择“设置主文件”命令，如图 5-71 所示。

## 2. 连编

应用系统的所有程序编写完成并设置了主文件后，即可进行连编。单击项目管理器右侧的“连编”按钮，在弹出的“连编选项”对话框中根据需要选择，如图 5-72 所示。

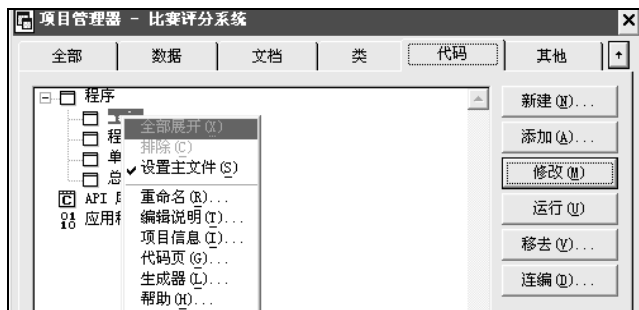


图 5-71 设置主文件



图 5-72 “连编选项”对话框



## 实施步骤

### 1. 设置主文件

右击选择项目管理器的“代码”选项卡下的 main.prg，在弹出的快捷菜单中选择“设置主文件”命令。

### 2. 连编

单击项目管理器右侧的“连编”按钮，参照图 5-72 选中“Win32 可执行程序/COM 服务程序”单选按钮，并勾选“重新编译全部文件”、“显示错误”、“连编后运行”3 个复选框，单击“确定”按钮。如果没有错误提示，系统将会生成“学生成绩管理信息系统.exe”文件。

## 任务二 发布“学生成绩管理信息系统”

### 任务目标

了解发布“学生成绩管理信息系统”等应用程序的方法和过程。

## 任务描述

将“学生成绩管理信息系统”项目文件进行打包发布。

## 相关知识

要将应用程序打包发布,需要使用 Visual Foxpro 9.0 安装盘中自带的 InstallShield 程序。安装 InstallShield 的方法是插入 Visual FoxPro CD,单击“Install InstallShield Express”按钮,再按照 InstallShield Express 安装向导指示进行操作。

安装 InstallShield 程序后,选择“开始”“程序”命令,即可打开“InstallShield Express”窗口进入程序。

打开 Start Page,如图 5-73 所示。

选择“File”“New”命令,弹出“New Project”对话框,选择“Express Project”选项,在“Project Name”文本框和“Location”下拉列表中,可以修改项目文件名和修改保存项目文件的位置,在“Project Language”下拉列表中选择“Chinese( Simple )”选项,勾选“Create project file in ‘Project Name’ subfolder”复选框,最后单击“OK”按钮,如图 5-74 所示,进入“project assistant”页面,如图 5-75 所示。

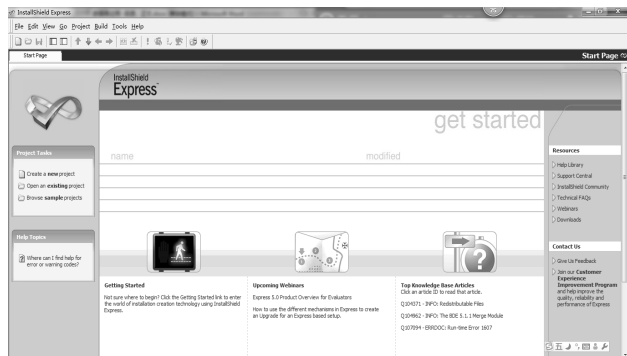


图 5-73 Start Page

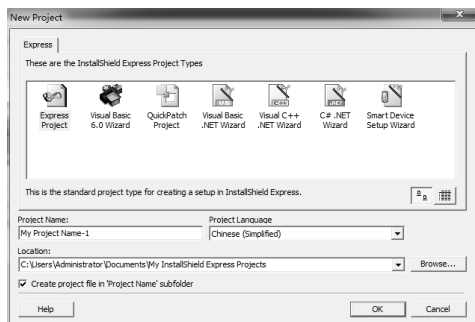


图 5-74 “New Project”对话框

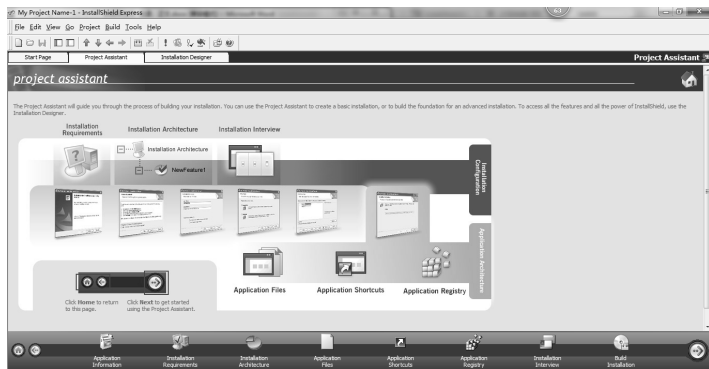


图 5-75 “project assistant”页面

项目辅助页中共包含 8 个操作步骤,用户可以根据实际需要对其中的全部或部分进行修改或设置。



Application Information（应用程序信息）：填写安装程序有关的信息，包括公司名称、安装程序名称、版本、网址及显示图标。

Installation Requirements（安装需求）：选择安装程序以后运行时所需要的操作系统，一般为默认选项即可。

Installation Architecture（安装架构）：一般选择默认选项。

Application Files（应用程序文件）：选择需要安装的文件或文件夹，选择 Add Files 加入。

Application Shortcuts（应用程序的快捷方式）：选择进入主程序的快捷启动方式。

Application Registry（应用程序注册）：对操作系统的注册表进行修改。

Installation Interview（安装概况）：主要对安装过程中出现的上述 ~ 项内容进行修改设置。

Build Installation（建立安装文件的类型）：如单独的可执行文件、网络安装、光盘安装等。

需要注意的是，此时先不要单击“Build Installation”按钮，而要先进入“Installation Designer”页面进行安装设计。

## 2. 打开页面

打开“Installation Designer”页面，如图 5-76 所示。

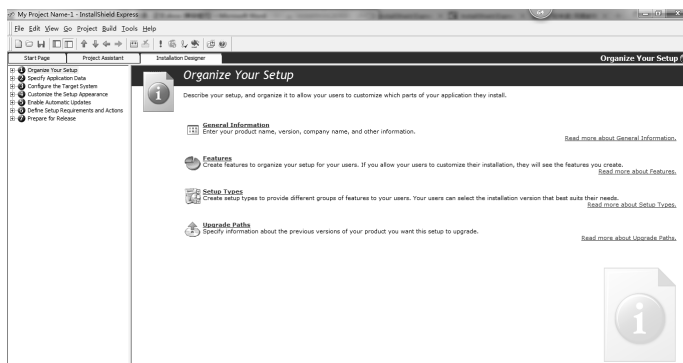


图 5-76 “Installation Designer”页面

在此页面中可以对“project assistant”页面中的设置进行修改，而且为了使应用程序能够脱离 Visual FoxPro 平台，需要将 Visual FoxPro 9.0 的实时运行库加入到安装文件中。选择页面左侧栏中第 项中的“Redistributes”选项，选中右侧 InstallShield Objects/Merge Modules 中的 Microsoft Visual Foxpro 9 Running Libraries，将其加入到安装文件中。

## 3. 完成安装程序的制作。

返回“project assistant”页面，在“Build Installation”中，选择独立文件、网络或光盘作为安装类型，单击“Build Installation”按钮，即可完成应用程序的打包程序。完成的安装文件位置会体现在 Output 窗口的最后一行蓝色文字中，把 setup.exe 作为程序的安装入口。最后可以将程序发布到需要的服务器。



## 阅读材料

文件的包含与排除

在发布应用程序时,有些文件是不能被用户更改的,有些文件则是允许用户更改的,这时就要对这些文件分类处理,允许用户更改的文件设置为排除,如果某些文件只允许用户读取而不能修改,那么要将这些文件设置为包含,使其被包含到系统中。

默认情况下,项目的程序文件,如表单、菜单、报表等都应设置为包含,而数据文件(如数据库、数据表等)应设置为排除,因为系统的界面一般不允许用户修改,而系统中的数据则可以随时添加或删除。

已经“排除”的文件在项目管理器中依然是可以看见的,但其前面会有特殊符号“⊗”。



## 项目小结

本项目介绍了应用程序发布的方法,这是应用程序的设计与编写完成后用户开始应用的重要环节。连编可以生成扩展名为“.exe”的可执行文件,使数据库应用系统脱离 Visual FoxPro 环境在 Windows 中独立运行,同时还可以将应用程序打包发布,使用户可以安装并使用应用程序。读者可以很容易地将这些技术应用到其他系统设计过程中。



## 上机实习

参照“学生成绩管理信息系统”的发布过程将“图书管理信息系统”发布出去,使其能够脱离 Visual FoxPro 环境而独立运行。



## 知识巩固

### 一、选择题

1. 在连编对话框中,不能生成的文件类型是( )。  
A. .dll                      B. .prg                      C. .exe                      D. .appP
2. 下面关于“主程序”的说法,错误的是( )。  
A. 主程序是整个应用程序的入口  
B. 一个项目管理器中只能有一个主程序文件  
C. 在 Visual FoxPro 系统中,只要是项目中的文件都可以设置为主程序  
D. 主程序的任务是设置应用程序的起始点、初始化环境等
3. 对 Visual FoxPro 9.0 的应用程序进行打包的工具是( )。  
A. Visual Studio                      B. Visual FoxPro 6.0  
C. SQL Server                      D. InstallShield

### 二、填空题

1. 连编程序时,首先应将某个程序设计为\_\_\_\_\_,使其成为应用系统中第一个执行的程序。
2. 使用 Visual FoxPro 设计的数据库应用系统,只能通过项目管理器中的\_\_\_\_\_命令才能最终生成可执行文件,使其脱离 Visual FoxPro 开发环境而独立运行。